

Copyright

by

Matthew Tierney MacMahon

2007

The Dissertation Committee for Matthew Tierney MacMahon
certifies that this is the approved version of the following dissertation:

Following Natural Language Route Instructions

Committee:

Benjamin J. Kuipers, Supervisor

Joydeep Ghosh

Jonas Kuhn

Dewayne E. Perry

Brian J. Stankiewicz

Following Natural Language Route Instructions

by

Matthew Tierney MacMahon, B.S., M.S.E.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2007

To my parents, Paul and B.J., for encouraging both wonder and accomplishment.

To my wife, Sarah, for her unflagging love, support, and understanding.

To all my friends, who have helped in innumerable ways.

Following Natural Language Route Instructions

Publication No. _____

Matthew Tierney MacMahon, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Benjamin J. Kuipers

Following natural language instructions requires transforming language into situated conditional procedures; robustly following instructions, despite the director's natural mistakes and omissions, requires the pragmatic combination of language, action, and domain knowledge. This dissertation demonstrates a software agent that parses, models and executes human-written natural language instructions to accomplish complex navigation tasks. We compare the performance against people following the same instructions. By selectively removing various syntactic, semantic, and pragmatic abilities, this work empirically measures how often these abilities are necessary to correctly navigate along extended routes through unknown, large-scale environments to novel destinations.

To study how route instructions are written and followed, this work presents a new corpus of 1520 free-form instructions from 30 directors for 252 routes in three virtual environments. 101 other people followed these instructions and rated them for quality, successfully reaching and identifying the destination on only approximately two-thirds of the trials. Our software agent, MARCO, followed the same instructions in the same environments with a success rate approaching human levels. Overall, instructions subjectively rated 4 or better of 6 comprise just over half of the corpus; MARCO

performs at 88% of human performance on these instructions. MARCO's performance was a strong predictor of human performance and ratings of individual instructions. Ablation experiments demonstrate that implicit procedures are crucial for following verbal instructions using an approach integrating language, knowledge and action. Other experiments measure the performance impact of linguistic, execution, and spatial abilities in successfully following natural language route instructions.

Acknowledgments

Like all large endeavors, this work would have never been accomplished without the support of others. Thanks to my wife, Sarah, who has supported me throughout the process, even when the end continued to slip from view. Thanks to my family, who supported me even when they didn't pretend to understand what I'm doing or why I'm doing it. Thanks to all of our friends who did the same.

Thanks to my mom, who helped me get started with computer science, typing in BASIC programs on our Commodore 64. Thanks to my dad, who encouraged me to reach for the stars.

My friends helped me throughout, even when I asked for yet another ride, room, or spot to leave my things. I'd like to thank all of the Lipsters, especially Austin and Lisa, Ryan and Renee, Karl and Cheryl, Dung and Andrea, Dave and Jiseon, Anuj, Daniel, Paul, and Stacy.

Thanks to JP, Kyler, Sahar, Jane, Matt, Matt, Amy, Travis, and Chaz for their thoughts on my work and for taking in a wayward computer geek in a Psych Lab. Thanks especially to Travis and Chaz for running subjects for me and to JP for programming for the Psych Lab, so I wouldn't have to.

Thanks to Pat, Joseph, Jeff, Aniket, Ram, Shilpa, Jonathon, and the rest of the Robotics lab for likewise taking in a wayward cognitive scientist who likes to play with robots.

Thanks to the great people I worked with at NASA JSC, especially those who still

ask how it's going and if I'll finish, especially Dave, Debbie, Scott, Pete, Eric, Rob, Jeff, Dan, and Rich. You got me hooked on reactive execution and human-robot interaction.

Thanks to the gang at NRL, who similarly support inter-disciplinary research on human-robot interaction, especially Alan, Magda, Dennis, Bill, and Sam. This project grew out of my summers there, working on GRACE and GEORGE. Thanks to all the dispersed, diverse Grand Challenge team members, too.

Thanks to the good folks at NASA Ames, for letting me play with their robots last summer and for their ideas and advice, especially Vandí, Ari, Hans, Michael, and Mike.

Thanks to all the great teachers I've had along the way, in Flower Mound, in Denton, in Palo Alto, in Berlin, in Vienna, in Houston, in Austin, in Washington, D.C., and in Mountain View.

Thanks to Joydeep, Jonas, and Dewayne for advice, pointers, and for listening and reading.

Thanks to Ben for asking for the impossible and to Brian for pointing out the possible and to both for supporting me throughout this strange quest, as mentors, as advisors, as bosses, and as friends.

Thanks to all the people I forgot to thank in the rush to finally finish.

Thanks most of all to everyone who went out of their way to help me through this, even with every choice I made to make things harder on myself.

This work was partially accomplished in three summer internships at the Naval Research Laboratory, in the Navy Center For Applied Research in Artificial Intelligence and one summer internship at NASA Ames Research Center, in the Control Agent Architectures and Intelligent Robotics Groups. The remainder of the work was accomplished at the University of Texas at Austin, in the Shape and Space Laboratory and the Intelligent Robotics Laboratory, in rented rooms in Austin, San Antonio, and Oakland, on planes in between, in coffee shops around the world, and hither and yon.

This work was supported by AFOSR grants FA9550-04-1-0236, FA9550-05-1-0321

and NIH grant EY016089 to Brian J. Stankiewicz, by NSF grant IIS-0413257 to Benjamin J. Kuipers, and by support for Matt MacMahon under ONR work order N0001405WX30001 for the NRL Research Option, Coordinated Teams of Autonomous Systems and from the California Space Grant Foundation.

MATTHEW TIERNEY MACMAHON

The University of Texas at Austin

August 2007

Contents

Abstract	v
Acknowledgments	vii
Contents	x
List of Tables	xvii
List of Figures	xix
Chapter 1 Introduction	1
1.1 Route Instruction Domain and Related Work	2
1.1.1 Route Instructions Understanding is Tractable	3
1.1.2 Do what I mean, not (just) what I said	3
1.1.3 Clear Evaluation and Comparison to People	4
1.1.4 Real-World Applications	4
1.2 Language and Task Corpus Methodology	5
1.2.1 General Definition of Language and Task Corpus Methodology	5
1.2.2 Application of Language and Task Corpus Methodology to Study Spatial Route Instructions	6
1.3 Human Route Instruction Studies	6
1.4 Resolving Linguistic Ambiguity by Task Execution	7

1.5	MARCO Architecture	8
1.5.1	Conceptual Structure of Instructions	8
1.6	MARCO Route Instruction Studies	10
1.6.1	Ablation Studies of Natural Route Instructions	10
1.7	Summary	13
Chapter 2 Related Work in Spatial Language and Instruction Following		14
2.1	Psychological Studies of Spatial Language	15
2.1.1	Spatial language and spatial cognition	15
2.2	Psychological Studies of Route Instructions	17
2.2.1	Gary Allen and colleagues	17
2.2.2	Michel Denis and colleagues	18
2.2.3	Barbara Tversky and colleagues	18
2.2.4	Other work	20
2.3	Route Instruction Generators	20
2.4	Automated Instruction Following and Analysis	21
2.4.1	Computational Models of Spatial Prepositions	21
2.4.2	Computational models of route instructions	22
2.4.3	GRACE and GEORGE in the AAI Robot Challenge	26
2.4.4	Instruction-Based Learning (IBL) Project	27
2.4.5	Spatial Ontologies and Spatial Instructions	27
Chapter 3 Language and Task Corpus Methodology		29
3.1	Language and Task Corpus Methodology for Spatial Route Instructions . . .	31
3.2	Related methodologies for evaluating natural language understanding systems	33
3.3	Related methodologies for studying natural language tasks	34
3.3.1	MAP-TASK Corpus Methodology	35

Chapter 4 Human Route Instruction Experiments	38
4.1 Motivating questions	39
4.2 Study overview and motivations	39
4.3 Apparatus	41
4.3.1 Control of movement through the environment	41
4.4 Stimuli	43
4.4.1 Environment Maps	43
4.4.2 Environment Landmarks	45
4.5 Human Directors Learn, Navigate, and Describe	46
4.5.1 Procedure	46
4.5.2 Route Instruction Entry	50
4.5.3 Experiment 1: Six directors across all three environments	54
4.5.4 Experiment 2: Twelve directors each in one environment	54
4.5.5 Experiment 3: Twelve directors with continuous motion control	55
4.6 Route Instruction Corpus Language Statistics	55
4.7 Human Followers Read, Navigate, and Evaluate	60
4.7.1 Procedure	60
4.7.2 Experiment 1: 24 people following 6 directors' instructions	62
4.7.3 Experiment 2: 44 people following 18 directors' instructions	63
4.7.4 Experiment 3: 24 people following 12 directors' instructions	64
4.8 Human Task Performance Overview	64
4.8.1 Differences in Directors	67
4.8.2 Differences in Human Followers between Corpus 1 and Corpus 2	70
4.8.3 Gender-linked performance differences	71
4.9 Discussion	73
Chapter 5 Spatial Route Instructions in the MARCO Architecture	76
5.1 Understanding and Following Route Instructions in Context	76

5.2	Syntax Parser	77
5.2.1	Training the Probabilistic Context-Free Grammar	79
5.2.2	Robustness for the Syntax Parser	79
5.3	Content Framer	80
5.3.1	Robustness to unknown words and sentence structures	81
5.4	Instruction Modeler	81
5.4.1	Representing Referring Phrases as View Descriptions	84
5.4.2	Representing Conditional Actions as Procedural Specifications	87
5.5	Executor: Interleaving Action and Perception	88
5.5.1	Inferring procedures implicit in instructions	90
5.5.2	Recognizing syntactic, semantic, pragmatic, and exploratory cues	91
5.5.3	Executing an Example from the Route Instruction Corpus	95
5.5.4	Other work on understanding implicit procedures	95
5.6	Robot Controller	96
5.7	View Description Matcher	97
5.8	Modeling Route Instructions in the HSSH Ontology	99
5.8.1	Relation to the Spatial Semantic Hierarchy	101
5.8.2	Modeling route instructions by topological maps	102
5.9	Extension to handle other sorts of ambiguity	106
Chapter 6 MARCO Route Instruction Corpus Experiments		109
6.1	MARCO Followers Model the Text and Navigate	110
6.1.1	Apparatus	110
6.1.2	Stimuli	111
6.1.3	Procedure	113
6.1.4	Evaluation	113
6.2	Full MARCO Performance	116

6.3	Comparing Implicit Procedures Inference to Fundamental Explicit Navigation Procedures	117
6.4	Implicit Procedures in Route Instructions	120
6.4.1	Implicit Procedure Cues Results by Rating	122
6.5	Inferring Different Types of Implicit Procedures	124
6.5.1	Implicit Procedure Experiment Results by Rating	126
6.6	Object, Structural and Appearance Landmarks	127
6.6.1	Review of Landmarks Types	127
6.6.2	Landmark Recognition Ablation Study	128
6.6.3	Landmark Recognition Ablation Results	130
6.7	Hybrid Spatial Semantic Hierarchy	132
6.7.1	Review of the Hybrid Spatial Semantic Hierarchy	132
6.7.2	Hybrid Spatial Semantic Hierarchy Ablation Experiment	134
6.7.3	Hybrid Spatial Semantic Hierarchy Ablation Results	134
6.7.4	Extensions to the Hybrid Spatial Semantic Hierarchy	138
6.8	Grammar Cross-validation	139
6.8.1	Review of Cross-Validation Methodology	139
6.8.2	MARCO Grammar Cross-Validation Experiments	140
6.8.3	MARCO Grammar Cross-Validation Results	140
6.8.4	Cross-Validation Discussion	144
6.9	Human-MARCO Discrepancy Analysis	145
6.10	Comparison to related work	148
6.10.1	Comparison to the Instruction-Based Learning project	149
6.11	Conclusions from MARCO experiments	151
Chapter 7	Conclusions	155
7.0.1	Future Work	156
7.1	Empirical examination of route instruction following	158

Appendix A Human Experiment Materials	160
A.1 Software configuration	160
A.2 Running the experiment	162
A.3 Example Consent Form	163
A.4 Director Guide	165
A.4.1 Exploration	165
A.4.2 Navigation Quiz	165
A.4.3 Giving Route Instructions	166
A.5 Director Key Meanings	168
A.6 Instruction Follower Guide	169
A.6.1 Movement Controls	169
A.6.2 Following Route Instructions	169
A.7 Follower Key Meanings	171
Appendix B MARCO Ablation Options	173
B.1 Options: Fundamentals	173
B.2 Options: Conditionals	174
B.3 Options: Heuristics	176
B.4 Options: Recoveries	178
B.5 Options: Tweaks	178
B.6 Options: Linguistics	179
B.7 Options: Implicits	182
B.8 Options: Landmarks	183
B.9 Options: HSSH	183
B.10 Options: Comparison	184
Appendix C Glossary and Language Model	185
C.1 Glossary	185

C.1.1	Route Instruction Analysis Vocabulary	185
C.1.2	Abbreviations	186
C.2	Route Instruction Grammar	186
C.2.1	Verbs	186
C.2.2	Nouns	187
C.2.3	Turn Command Arguments	188
C.2.4	Travel Command Arguments	189
C.2.5	Description Utterance Arguments	190
C.2.6	Adjectives	191
C.2.7	Adjectival Phrases	191
C.3	Interfaces	192
C.3.1	Simulation	192
C.4	Representation of Procedural Specifications	193
C.5	Representation of View Description	194
	Bibliography	196
	Vita	227

List of Tables

4.1	Characteristics of the three testing environments.	44
4.2	Statistics per group of instructions by each director.	56
4.3	Corpora text statistics averaged per director group	57
4.4	Most frequent words per group of instructions	58
4.5	Most distinctive words per group of instructions.	59
6.1	Comparing performance of MARCO to people and to MARCO following only explicitly commanded procedures.	116
6.2	Comparing performance of MARCO on inferring implicit proceures and executing types explicitly commanded proceures	119
6.3	Comparing performance of MARCO versions without different implicit procedure cues	123
6.4	Comparing the performance of MARCO without implicit turns and travels .	124
6.5	Comparing the performance of MARCO without the ability to recognize different kinds of landmarks	129
6.6	Performance across subjective ratings ablating representations from the HSSH136	
6.7	Performance on cross-validation runs by subjective rating.	141
6.8	Discrepancy analysis for sampled instructions from Corpus 1.	146
6.9	Discrepancy analysis for sampled instructions from Corpus 2.	153
6.10	Discrepancy analysis for sampled instructions from Corpus 3.	154

A.1 Combinations for director experiments 2 (Corpus 2) and 3 (Corpus 3). . . . 161

List of Figures

4.1	Sample views of all objects, from the <i>Medium</i> environment.	42
4.2	Sample views of all hallway textures, from the <i>Compact</i> environment. . .	47
4.3	Example of route instruction window, with entered text.	51
4.4	Dialog box for rating route instructions.	52
4.5	Dialog box for rating navigation confidence.	53
4.6	Occurrence and success rates over all human followers by mean rating. . . .	65
4.7	Mean performance over all human followers	66
4.8	Success rates over all human followers per Corpus 1 director.	67
4.9	Success rates over all human followers per Corpus 2 director.	68
4.10	Success rates over all human followers per Corpus 3 director.	69
4.11	Success rates for all human followers from each experiment by instructions rating.	70
4.12	Success rates for human followers and directors by gender.	72
5.1	MARCO linguistic modules modeling a route instruction text	78
5.2	How interpreting an utterance depends on the follower's knowledge of its pose in the environment.	108
6.1	Human visual view and MARCO symbolic view of a hallway.	111
6.2	Human visual view and MARCO symbolic view of a shorter hallway	112

6.3	Human and full MARCO success rates versus <i>post hoc</i> human ratings. . . .	115
6.4	Comparing inferring implicit procedures vs. executing types of explicit procedures	118
6.5	Performance for people, MARCO and MARCO without implicit procedure cues	121
6.6	Implicit turns have a higher impact than implicit travels.	125
6.7	Success rates for MARCO without the ability to recognize types of landmarks.	130
6.8	Performance graph ablating representations from the Hybrid Spatial Semantic Hierarchy	135
6.9	Success rates for MARCO under cross-validation across instruction quality.	142
6.10	Success rates for MARCO under cross-validation per director.	143
A.1	Vizard Experiment Starting Dialog	172

Chapter 1

Introduction

Since Turing's seminal paper (1950), a natural response to language has been regarded as the key test of intelligence. Full and rich language use is perhaps the most defining characteristic of human intelligence. However, despite more than a half-century of trying, the goal of natural language interaction with an artificial partner remains distant.

One of the most practical applications of natural language is following verbal instructions. With instructions, one person, the *director*, uses language to guide others, the *followers*, in accomplishing complex tasks when the followers do not have the knowledge, expertise, authority, or time to plan. Let us distinguish *instructions* as describing complex procedures of multiple actions, while a *command* impels one procedure. Instructions are ubiquitous across human tasks. Common applications include recipes, assembly instructions, spatial route instructions, repair manuals, tutoring, coaching, as well as innumerable unnamed, *ad hoc* interactions where one person guides another through a procedure.

For perfectly clear, explicit, and correct instructions, the instruction follower can directly map the syntax and surface semantics of *what was said* into an imperative model of *what to do*. To handle instructions that are partially unclear, implicit, or incorrect, the follower must achieve a deeper understanding of the pragmatics of the instructions; that is,

what the director meant the follower to do. How often instructions fail to be clear, complete, or correct is an empirical question that must be answered to understand the difficulty of following route instructions.

This dissertation aims to provide a computational model of route instruction following with accuracy and robustness comparable to human performance, despite natural mistakes, omissions, and variation in the instructions. Five questions must be answered to build and evaluate a natural language understanding system that follows instructions: (1) How do directors naturally describe the task in instructions? (2) How do human followers act when following these instructions – particularly, how often do people succeed at the task given the instructions? (3) What challenges arise in following instructions and how can they be robustly handled? (4) What computational principles are necessary to follow route instructions, especially to handle under-specification and ambiguity? (5) How much does each system component impact the system’s performance, measured by how often the system successfully follows natural instructions?

1.1 Route Instruction Domain and Related Work

Verbal *route instructions* are a type of instructions where the *director* intends to guide a mobile agent, the *follower*, to a specific spatial destination. When following route instructions, the follower must parse and interpret the text, model the instruction’s actions and descriptions, and perform perceptive and movement actions to navigate to the destination. Correctly following instructions requires executing both explicitly commanded procedures and procedures implicit in the instruction language.

Route instructions make a compelling domain for several reasons. Instructions require integration of language, action, and reasoning skills, so draw on research in linguistics, psychology, and artificial intelligence. Route instructions are constrained enough to be tractable to approach, while being complex enough to be interesting. The evaluation of following route instructions is clear – whether the follower ends at the goal.

Finally, spatial route instructions have compelling real-world applications.

1.1.1 Route Instructions Understanding is Tractable

Spatial route instructions have several distinct features within the space of instructions that make building a follower tractable. First, there are well described theories of acting, reasoning and talking about large-scale spaces (Kuipers et al., 2004; Kuipers, 2000; Landau and Jackendoff, 1993; Siegel and White, 1975; Timpf et al., 1992; Yeap and Jefferies, 1999). Second, moving through large scale space requires just two fundamental spatial actions, turns and travels, reducing the number of verbs to model. For the pure task of route following, few manner verbs, which specify how to perform actions, are necessary. Moving through space is inherently sequential, and the structure of route instructions tends to also be highly sequential. Some other types of instructions, such as recipes, may involve executing many extended actions in parallel. Finally, as discussed below, route instructions provide a clear and unambiguous evaluation metric. With multiple people following each set of instructions, or measuring the efficiency of navigation, route instructions also allow measuring a gradation of success.

1.1.2 Do what I mean, not (just) what I said

Route instructions are not trivial, however. Route instructions allow a lot of variation, in which landmarks and route attributes described and how they are verbalized. When more than one route is possible, the directors can also describe different procedures that will accomplish the task. Route instructions require more than mindlessly executing the explicit instructions when they are ambiguous, erroneous, or contain implicit procedures. The follower must determine which necessary, but unstated, procedures satisfy the stated or implicit preconditions. The follower must apply a knowledge of large-scale space and knowledge of linguistic conventions in route instructions in order to succeed, despite the challenges of natural instructions.

1.1.3 Clear Evaluation and Comparison to People

Route instructions provide a clear evaluation metric: *Does the follower understand the instructions well enough to arrive at the destination?* The task of navigating along a complex route through a large-scale, unknown space is a sequential decision-making problem. If any mistake or omission – by either the director or follower – is not corrected, the follower will end up at the wrong destination. This distinguishes the task from other task-centered language studies, where people can correctly finish at a fuzzy distribution of places (Roy, 2005; Skubic et al., 2004b; Tellex and Roy, 2006), where the result of text understanding is text retrieval or generation (Manning and Schütze, 1999; Reiter and Dale, 1997), or where the success criterion is unclear (Anderson, 1984; Anderson et al., 1991; Carletta and Mellish, 1996; Levit and Roy, 2007).

The situated context of instruction-directed task execution gives an empirical view into when inference is required. There is a long history looking at drawing the pragmatic meaning from a text – what is entailed by it or implicated by what was said (Grice, 1967, 1975, 1989; Grodner and Sedivy, 2004; Perrault and Allen, 1980; Sedivy, 2003; Sperber and Wilson, 1986; Stone et al., 2003). However, this leaves open the question of how often these implicatures and entailments are necessary to understand the text, for instance, to follow instructions successfully. Natural language under-specifies descriptions; route instructions are no exception. Only some aspects of some landmarks and some features of some actions along the route are described. The number of conceivable implicatures and entailments from a set of route instructions is large, if not infinite. By comparing the performance of software agents to human performance, we can discover *which* inferences are necessary to follow the instructions and *how often* people make them.

1.1.4 Real-World Applications

Route instructions are potentially useful anywhere a person is interacting with a mobile robot in a complex environment. Route instructions can be used as an assistive technology,

to ease (or allow) the control of smart wheelchairs (Beeson et al., 2007; Simpson, 2005; Tellex and Roy, 2006). Route instructions could help in Urban Search and Rescue, to guide a rescue or exploratory robot through an unknown environment even if communication is lost (Burke et al., 2004; Murphy, 2004). Route instructions would also assist astronauts supervising semi-autonomous robots (Burridge et al., 2003; Fong and Nourbakhsh, 2005; Kortenkamp et al., 1998).

1.2 Language and Task Corpus Methodology

This thesis looks at the problem of following *natural instructions*. Let us use the term *natural instructions* to refer to instructions that have the characteristics of naturally occurring instructions: instructions that are free-form, natural language with unconstrained vocabulary or grammar and containing naturally occurring mistakes and omissions. Specifically, to elicit natural instructions, directors learned the environment and navigation tasks through first-person experience, recalled the environment from memory while planning the task, and gave the instructions to unknown followers with minimal knowledge of the environments. Each of these factors increases the ecological validity of the instruction corpus (Cohen, 1995; Rosenthal and Rosnow, 1991).

1.2.1 General Definition of Language and Task Corpus Methodology

The first general contribution of this dissertation is a methodology for building software systems that follow free-form, natural language instructions. First, gather a corpus of people giving instructions to accomplish concrete tasks in the domain. Second, observe other people following and rating these instructions to accomplish the task. Third, build a working skeletal system able to interpret and execute the most common instructions in the domain. Comparing the difference between people's performance in following the instructions with the system's performance shows where to focus development effort, as well as providing regression testing. Once the system is approaching human performance,

we can measure the impact of each component in the system – or each feature in the instructions – by measuring the performance of the system without that component. Note that while this work focuses on spatial route instructions, the language and task corpus methodology is general for any procedural instructions.

1.2.2 Application of Language and Task Corpus Methodology to Study Spatial Route Instructions

This dissertation applies the language and task corpus methodology to spatial route instructions, implementing a system that follows spatial route instructions through an unknown large-scale space. By ablating components from the route instruction following agent and observing their impact on performance, we measure the importance of general instruction following abilities, domain-specific navigation and spatial reasoning abilities, and the combination of both general and task-specific abilities. These computational experiments reveal which skills are necessary to follow natural instructions and how often each is required.

1.3 Human Route Instruction Studies

We gathered a large corpus of route instructions from three human experiments, using the same apparatus, stimuli and per-trial procedure. In the first study, six directors each gave instructions for 42 routes between named positions in three virtual environments. These instructions were used to develop a software route instruction follower, MARCO. In the second study, an evaluation corpus was gathered, where 24 directors each gave 42 instructions in one of the three environments. This study also varied the start and end points of the requested routes and tested both continuous and discrete movement through the environments. The second half of the corpus is used to evaluate the generality of MARCO’s methods across individual differences in linguistic and spatial reasoning for route instructions.

1.4 Resolving Linguistic Ambiguity by Task Execution

The challenges in instruction following, and the methods for handling them, occur at every level of the computational architecture. At the lexical level, the system should be robust to unknown words. At the syntax level, it should handle the variety of utterance forms that people can understand, whether or not formally grammatical. At the semantic level, the system extracts the director's intended meaning across differences in surface form. The semantic model of the instructions captures the both imperative and declarative constraints in the instructions – both how the follower should act and what the follower should expect to observe. At the pragmatic level, the agent infers which procedures are required, whether they are left implicit – or even stated incorrectly – in the route instructions.

We have implemented these methods as an integrated system, MARCO, and we have shown that they achieve comparable levels of performance to human followers, across the range of quality of human-provided route instructions. The architecture is described in detail in Chapter 5, and the individual challenges and methods are described in Chapter 6. The performance of the full agent approaches the performance of people following the instructions overall; for the best-rated instructions, MARCO performs nearly at human levels. This empirical validation supports the hypothesis that the MARCO architecture captures the challenges in spatial route instruction following and the methods that humans use to meet those challenges.

The major conceptual contribution of the MARCO architecture is deferred handling of ambiguity. Both referring phrase resolution and modeling instructions as a sequence of procedures introduce semantic ambiguity. Referring phrases vastly under-specify the configuration, even combined with other knowledge explicit in the instructions or in unspoken shared common heuristics – common-sense. A paragraph of instructions vastly under-specifies the procedures necessary to accomplish the described task. Even a single instruction command can refer to a complex conditional procedure (Bugmann et al., 2004; Lauria et al., 2002a; Simmons et al., 2003; Tellex and Roy, 2006).

MARCO handles what Sperber and Wilson (1986, 2004) called “contextual implication,” conclusions derivable from both the linguistic input and the external context, but from neither alone. MARCO defers handling some linguistic ambiguity until it is situated at that point in the task and environment, when it has more perceptual and cognitive common ground with the director, by perceiving the environment at that point in the route.

1.5 MARCO Architecture

This dissertation details a system, MARCO, that interprets human free-form route instructions and follows the inferred model of the described route (MacMahon et al., 2006). This work builds on a rich literature studying different aspects of route instructions. Some work presents a model of route instructions, but does not apply the model to navigate (Anderson et al., 1991; Daniel et al., 2003; Denis et al., 1999; Klippel et al., 2005; Tversky and Lee, 1999; Vanetti and Allen, 1988). Other work concentrates on understanding single spatial commands in the small-scale space of a room (Skubic et al., 2004b; Tellex and Roy, 2006) or tabletop (Roy, 2005). Finally, other work follows instruction sequences in a large-scale space, but does not use spatial and linguistic knowledge to recover from instruction errors or to infer implicit procedures (Bugmann et al., 2004; Simmons et al., 2003).

MARCO is composed of six primary modules: three to interpret the route instruction text linguistically and three to interpret and execute the instructions in the context of the task and environment. MARCO’s general framework is domain-independent, although extending the architecture to domains outside of large-scale spatial route instructions, such as telephone help or booking systems, remains future work.

1.5.1 Conceptual Structure of Instructions

MARCO’s instruction modeler produces two interlinked models: an imperative, procedural model of the actions to be taken – a skeletal plan for the task – and declarative models of the expected environmental and task states. MARCO models instructions as a series

of parametrized local procedures, called *procedural specifications*. Each procedure in the instructions is modeled as a procedural specification. Some specified conditions of procedures are modeled either as internal state, such as estimates of distance traveled. External state is modeled as *view descriptions*, which constrain what the follower expects to see, given the referring phrases and commands in the route instructions.

The procedures to fulfill each condition are modeled as embedded procedural specifications. A command to travel down a hallway is labeled *Travel^p*. *Travel^p* is composed of the explicit, simple causal *travel^a* action to reach the next place, but also may require other actions such as *turn^a* and *verify^a*. For instance, “Take the blue hall to the chair,” may require a *travel^a* actions to move to the blue hall, a *turn^a* actions to face along the blue hall towards the chair, and the explicit *travel^a* actions along the blue hall until the chair is reached. However, since the number of actions to execute is not known *a priori*, the possible procedure sequences are represented as reactive procedures, which are invoked as necessary. The first *Travel^p* procedure to the blue hall may, in turn, require a *Turn^p* procedure to face the blue hall and possibly a *Find^p* procedure to bring the blue hall into view.

The expected observations of the blue hall and chair in this example are modeled as view descriptions, which model the required relative distance and position as well as the landmarks’ type and appearance. For instance, at the beginning of the main *Travel^p* down the blue hallway, there should be some sort of blue path directly in front of the follower, and a chair in the distance on the part of that path in front of the follower.

The instruction modeler also decomposes high-level commands into lower-level procedures. The concise “Take the third right to the end of the hall,” is modeled and executed no differently than if the director had explicitly commanded, “Go down one path to the third place with a path to the right. Turn right there. Go down that path until you reach the end of that hall.” This simplifies the execution code, by separating it from the surface form of the instructions.

Though the implemented MARCO agent only follows route instructions through large scale spaces, these knowledge representations are general to modeling complex reactive procedures. Some of the modeling methods and heuristics are specific to this domain, but many are generalizable.

1.6 MARCO Route Instruction Studies

MARCO follows the same instruction texts to navigate through the same environments as people do. This allows direct comparisons of MARCO, MARCO with abilities ablated, and people, each following the same instructions. For people, the results are the mean over runs from multiple participants following each instruction set, each beginning at the start location facing a random direction. For the MARCO cases, the results are the mean over runs facing each of the four directions at the start. MARCO's input was from the hand-verified 'gold-standard' parse treebank, not the parser, but all other modeling was done autonomously.

1.6.1 Ablation Studies of Natural Route Instructions

The implementation of MARCO is configurable to easily remove some capabilities at runtime. This enables computational experiments of running versions of MARCO with different language, action, perception, and reasoning abilities on the same instruction corpus. These experiments give insight into how people give and follow instructions, and how important these various capabilities are to construct a software agent to follow the instructions.

How important are implicit procedures?

One key finding is the importance of executing the implicit procedures in the instructions, not just those that are explicitly commanded. Executing only explicit procedures, MARCO succeeded on just 34% of the trials.

What are the cues for implicit procedures?

Instructions can imply unstated, implicit procedures with four distinct cues. First, *syntactic cues* are domain independent syntax that mark an explicit condition which the follower must take unspecified procedures to achieve. Second, *semantic cues* require the decomposition of high-level procedures into low-level actions and task conditions. Third, *pragmatic cues* alter the interpretation of commands and descriptions, depending on their surrounding context in the instructions and in task execution. Fourth, *exploratory procedures* are taken to gain information needed to match referring phrases to the task and environment state.

These four triggers of implicit procedures occur in instructions across all domains, though this work examines them in the context of spatial route instructions. This section compares the results of MARCO running without the ability to recognize different kinds of cues for implicit procedures: cues implicit in the syntax, verb semantics, or discourse and utterance pragmatics, and exploratory procedures used to gain information. Without semantic cues, MARCO follows only 36% of the instruction corpus; MARCO without semantic cues, 40% success rate; MARCO without exploration procedures, 50%; and MARCO without pragmatic implicit procedures, 55%.

What kinds of landmarks are necessary?

Previous work examined the role of object and structural landmarks in learning and navigating large-scale spaces (Stankiewicz and Kalia, 2007). A large body of related work has examined the role of landmarks in spatial route instructions (Daniel and Denis, 2004; Klippel and Winter, 2005; Lovelace et al., 1999; Michon and Denis, 2001; Nothegger et al., 2004; Raubal and Winter, 2002; Weissensteiner and Winter, 2004). We can measure how often structural and object landmarks are mentioned in route instructions by examining statistics available from our language modeling. We can examine when the landmarks are crucial for navigation, as opposed to used in elaboration, by selectively removing the ability to recognize these different types of landmarks. Selectively removing the

ability to recognize different types of landmarks reveals that without Object Landmarks, MARCO succeeds on 49% of the corpus; without Intersection Landmarks, 48%; without Appearance Landmarks, 44%; without Causal Landmarks (paths and walls), 34%; and without Structural Landmarks, 27%.

What spatial cognition is necessary?

Following spatial route instructions requires the ability to represent and manipulate space in several different ontologies (Beeson et al., 2007; Kuipers et al., 2004; Kuipers, 2000), as well as different skills at each level. The *Control* level models spatial actions as control laws, requiring skills such as moving until a condition is met (closed loop control) and moving an estimated distance or turning an estimated angle (open loop control). The *Local Metrical* level models the local environment geometrically. Skills at this level include maintaining the relative position of landmarks that are no longer visible and *perspective taking* (Trafton et al., 2005; Tversky et al., 1999), the ability to reason about perspectives other than the currently view. The *Local Topological* level models the navigational affordances of the local space, i.e. what sort of intersection it is, if any. At the *Causal* level, space is represented by how abstracted views are linked by abstracted actions (applications of control laws). The *Global Topological* level reasons about space as networks of places and paths.

By selectively ablating spatial abilities in MARCO and measuring when the agent can no longer reach the destination, we learn when these skills are necessary to follow route instructions in large-scale spaces. These experiments reveal how often the diverse spatial reasoning and representation skills play crucial or elaboration roles in natural human route instructions. Without the Local Topological level, MARCO follows 47%; without the Topological level, 47%; without Open Loop Causal procedures, 25%, without the Local Metrical level, 17%; without Closed Loop Causal procedures, 7%; and without any Causal procedures only 1% of the instruction corpus.

1.7 Summary

This dissertation describes a general architecture for following natural language instructions, with an implementation applied to following spatial route instructions through an unknown large-scale space. A new, large language and task corpus was collected, of human directors' exploration traces and route instructions; and human followers' navigation traces using these instructions and subjective ratings. The performance of the full agent approaches the performance of people following the instructions overall and is statistically equivalent to the performance for the best-rated instructions. By selectively removing language, action, perception, and spatial reasoning abilities, the evaluation measured the importance of each ability for following spatial route instructions.

Chapter 2

Related Work in Spatial Language and Instruction Following

Instructions in general require applying an inter-disciplinary understanding of natural language and acting in the world. An integrated cognitive system is required to correctly follow free-form, natural language instructions. Because route instructions require many different abilities to follow, they have been studied across the cognitive sciences. In cognitive psychology, route instructions show how people think about space (Anderson et al., 1991; Daniel et al., 2003; Denis et al., 1999; Lovelace et al., 1999; Tversky and Lee, 1999; Vanetti and Allen, 1988). In linguistics, route instructions show how people talk about space (Edmonds, 1993, 1994; Levelt, 1982; van der Zee and Slack, 2003). In artificial intelligence, spatial commands and route instructions are studied for natural human-robot interaction (Bugmann et al., 2004; Klippel et al., 2005; Simmons et al., 2003; Skubic et al., 2004b; Tellex and Roy, 2006). Successfully following route instructions requires integrating techniques from all of these fields.

2.1 Psychological Studies of Spatial Language

2.1.1 Spatial language and spatial cognition

Route instructions are a sequence of descriptions of spatial actions and configurations. Spatial cognition is a fundamental cognitive skill: across tasks and languages, linguistic and psychological studies have found strong evidence of a “Where” system that recognizes and verbalizes spatial relations separately from object geometry (the “What” system) (Jackendoff, 1983; Landau and Jackendoff, 1993). Furthermore, when describing a route, people express similar information whether communicating in language (verbal instructions) or in pictures (maps) (Hayward and Tarr, 1995; Klippel et al., 2003; Tappe and Habel, 1998; Tversky and Lee, 1999).

Route instructions are an interesting case of a more general class of problems, verbally describing a spatial environment. There has been a wealth of good work in studying the different aspects of describing spatial relations, layouts, and scenes. Herskovits (1997) provides an invaluable survey of language about space. Mukerjee (1998) surveys cognitive representations of space, and in particular which are quantitative (“neat”) and which are qualitative (“scruffy”). van der Zee and Slack (2003) recently collected essays surveying work on directional prepositions.

Linde and Labov (1975) focus on one domain particular of describing large-scale space. This work presents a grammar for the sentence- and discourse-structure of verbal descriptions of apartments by their residents. Linde and Labov derive twelve rules of the grammar, delineating what is described, the order of the description, how sentences or subordinate clauses are formed, and the transition from action verbs to passive spatial layout descriptions. This engrossing paper describes both the semantics and pragmatics (“well-formedness”) of large-scale space descriptions. The paper illustrates the derivation and application of the rules with examples of the collected apartment layout descriptions. The paper is a summary of (Linde, 1974).

Another keystone work in this field is Landau and Jackendoff (1993): “‘What’ and ‘Where’ in Spatial Language and Spatial Cognition.” Landau and Jackendoff survey a variety of linguistic and psychological evidence supporting independent mental representations and linguistic structures for describing objects and locations. Named objects specify detailed geometric characteristics, especially shape. When referring to spatial relations and locations, however, the implicitly described geometry is more qualitative and vague. These distinct manners of reference to objects and locations, universal across human languages, mirror psychological and neurological evidence for separate modules: one reasoning about identifying an object and the other reasoning about spatial relations.

The language used to describe space naturally parallels the underlying representations. Talmy (2000) focuses on the closed sets of prepositions that we primarily use to describe spatial relations. A spatial preposition in a language encodes a rich set of default attributes of the relation it is describing. For instance, if one says object *A* was *across* object *B*, we generally infer that *A* begins on one side of *B* and extends continuously past the other side on a fairly straight path. *A* and *B* are both likely significantly longer in one dimension than the other and the long axes should run approximately perpendicular to one another, and *A* is likely shorter than *B*. We would make significantly different assumptions if we heard that *A* was *along*, *over*, *on*, *in*, or *around* *B*. The complete set of these default attribute values may not be true for any given relation, but if too many are violated, the preposition was the wrong word to describe the situation.

The MAP-TASK corpus is another influential study of spatial language. See Section 3.3.1 for a comparison of the MAP-TASK with our language and task corpus methodology.

2.2 Psychological Studies of Route Instructions

2.2.1 Gary Allen and colleagues

One of the most cited psychological studies of route instructions is Vanetti and Allen (1988). Vanetti and Allen looked for differences among subjects divided into four even groups by standardized testing: high spatial-high verbal, high spatial-low verbal, low spatial-high verbal, and low spatial-low verbal. Interestingly, the two standardized tests of spatial ability measured small-scale spatial ability, but were good predictors of large-scale space route planning ability. The subjects gave spoken verbal route instructions between two known buildings on their college campus and followed spoken verbal route instructions between two offices inside a campus building. The route across campus was “familiar to all subjects and not extensive.”

Spatial ability had a larger effect in the accuracy of subjects’ described routes than verbal ability. However, subjects with high verbal ability were more likely to describe the key choice and termination points. The subjects’ route instructions were not empirically tested by having others follow them, but by the experimenters coding them.

Allen (2000) later aims at capturing the “best-practices” of real-world direction giving. Allen summarizes experiments suggesting descriptives and delimiters should be inserted at choice points instead of en-route. Allen finds men are more persistent in following instructions; in his experiments, men have fewer points where they claim the route instructions were insufficient than women. Women’s performance improves when environmental features (landmarks, relative spatial directions) are emphasized over metrical distances and cardinal directions, but still perform worse than men. Allen also finds the quality of instructions is more important as the follower nears the destination, rather than the beginning of the route.

Allen (2003) followed up with a study of how and when gestures accompany route instructions. Allen found deictic gestures were the most frequent, especially emphasizing

right or left on turns. Iconic and jabbing emphasis gestures were less common than deictic gestures and gestures were more common in general with rapid speech.

2.2.2 Michel Denis and colleagues

Denis (1997) breaks down route instruction-giving into three phases: activating relevant spatial knowledge, determining a route, and translating that route into a verbal output. Denis codes a set of route instructions by breaking down each utterance into “minimal units of information.” Daniel et al. (2003) found good, poor, and “skeletal” instructions were differentiated by whether the proper action was associated with the proper landmark.

Fontaine and Denis (1999) followed up by examining how people give route instructions for three-dimensional routes through the Paris Metro differ. They found that underground, people specified actions in relation to objects, especially signs, far more frequently than when in open outdoor spaces. Michon and Denis (2001) guided subjects on long, but topologically simple routes through Paris districts. The subjects then repeated the route on their own and gave route instructions for a tape recorder. This experiment found that landmarks, while mentioned all along a route, are most frequently mentioned “close to critical nodes,” e.g. around a critical turn or picking the correct street to exit a large square. Recent work elaborates on these themes (Denis et al., 1999; Mellet et al., 2000), for instance, Daniel and Denis (2004); Daniel et al. (2003) found “good”, “poor”, and “skeletal” instructions were differentiated by whether the proper action was associated with the proper landmark.

2.2.3 Barbara Tversky and colleagues

Tversky and Lee performed a series of studies on how people direct others on routes in the different modalities of route instructions and sketch maps. Students were asked to give route instructions from a campus landmark to a well-known restaurant in a neighboring town. One work, Tversky and Lee (1998), follows Denis (1997) in describing each segment of a

route by its starting point, re-orientations, path progression, and a goal description. “How Space Structures Language” likewise follows up on Talmy (1983, 2000). They find support for Talmy’s schematization of space in the similar ways people describe routes verbally and pictorially. For instance, both verbal route instructions and pictorial route maps schematize information about a route into a series of generic turn and travel actions. In language, turns are represented by a couple of verbs or phrases, such as “turn,” “make a” and “take a”. In diagrams, turns are represented as arrows or sketches of intersections with orthogonal angles.

Tversky and Lee (1999) find the same roles are played by elements in a sketch map and hypothesize that there may be a common cognitive representation that underlies the generation of each. This work was supported by further work in Tversky (2000). Agrawala and Stolte (2001) implemented principles from this work in a software system that rendered sketch-style route maps. Their evaluation found people preferred the sketch maps over or in addition to more accurate and detailed cartographic maps.

Taylor et al. (2001) look at when and why people switch perspective while describing environments and routes. Taylor and Tversky (1996) examine the additional ambiguities introduced by describing an environment using a linear, limited natural language as compared to using analog depictions.

Another angle of Tversky’s research has been investigating the role perspective plays in spatial description and route instructions (Tversky and Lee, 1999). Taylor and Tversky (1992) looked at the differences in mental representations of large-scale space that resulted from differences in the perspective of a spatial description. Subjects read descriptions written either from a survey perspective or as a narrative of a route.

Tversky et al. (1999) examine the cognitive costs and benefits of changing perspective during verbal spatial description. They review evidence of costs in terms of both effort and errors when readers are forced to change perspective. The authors propose that speakers and writers may switch perspectives to take advantage of relatively more salient

objects and spatial relationships that are easier to describe and compute. The other proposed explanation is simply that people's heterogeneous mental representations of space encode different perspectives.

2.2.4 Other work

One of the earliest studies of route instructions was by Elliot and Lesk (1982). Edmonds (1993, 1994) examined reference resolution to previously unknown objects encountered in route instruction texts.

Lovelace et al. (1999) had college freshmen write route instructions across their college campus. Subjects first described two previously known routes while in a lab setting from a familiar part of campus. Twice, the students were led along an unknown route on campus, then described it. The route instructions were rated and coded for mentioning certain features, such as when and where landmarks are mentioned. Subjects often omitted mentioning turns, short segments, and landmarks, especially when travel was constrained by environmental considerations. Good route instructions mentioned many landmarks along the paths, off the route, and at the choice points, in contrast to other studies.

Buhl (2003) also looked at the effect of perspective (called "speaker orientation") on route instructions. Subjects gave route instructions to a listener with a different point of view and found subjects most often produced route instructions composed from their own perspective

2.3 Route Instruction Generators

Davis (1986) implemented the "Back-Seat Driver" system, an early in-car navigation system, producing real-time route instructions for a driver.

Moulin and Kettani (1998)'s GRAAD software generates a logical, specification of a route from a "Spatial Conceptual Map" and tests them by giving them to a virtual pedestrian in a simulated environment. This logical formulation is processed by another

module to convert it into natural language by removing redundant information, matching logical terms with environment names and matching logical relations with verbs. Gryl et al. (2002) later presented a richer conceptual model of English and French spatial expressions. Porzel et al. (2002) examine issues of how to linearize a representation of a two- or three-dimensional environment or scene into a one-dimensional string of words.

Fraczak et al. (1998) examines automatically generating route instructions in underground, three-dimensional environments, such as subway stations. Skubic et al. (2001) generated spatial descriptions of small-scale space for a robot navigating within a room.

Stocky (2002) implemented a kiosk system with a virtual avatar that used gesture and natural language route instructions to guide visitors to offices. From a hand-coded map, Stocky's software, MACK, generated spoken route instructions coordinated with the avatar pointing and highlighting a map. MACK also reasons about when to shift the perspective of route instruction-giving, based on Taylor and Tversky (1996). Kopp et al. (2007); Striegnitz et al. (2005) continue to study generating route instructions in both text and gestures.

2.4 Automated Instruction Following and Analysis

2.4.1 Computational Models of Spatial Prepositions

Several software systems have implemented computational models of spatial prepositions.

Winograd (1972) had one of the first implementations, with his SHRDLU system. SHRDLU executed single commands in a "Blocks World" domain, including planning to achieve commands with complex unsatisfied preconditions. SHRDLU performed using a controlled vocabulary of about fifty words, although it could learn nouns online. SHRDLU had models of the preconditions of prepositions such as *on*.

André et al. (1986) implemented a dialogue system that could find the reference objects for certain (German) spatial prepositions in a dialogue about a sightseeing

in a simulated city. Their system, CITYTOUR, could handle several basic and hedged prepositions to answer questions such as (translated to English) “Is the post office beside the church?” Regier and Carlson (2001) and Coventry and Garrod (2004) present implementations of system that ground knowledge of small-scale spatial prepositions by modeling both geometric and causal relations between objects.

Blisard, Skubic, and colleagues implemented a spatial referencing system on a mobile robot that can understand small-scale spatial prepositions such as *front*, *left*, *right*, *behind* (Blisard and Skubic, 2005; Blisard et al., 2006; Skubic et al., 2004b). Their system can ground prepositional phrases using these spatial prepositions to a occupancy grid representation of the immediate space surrounding the robot. Their implementation can describe the locations of objects surrounding the robot, can answer simple questions, and can move to achieve single commands to move to locations in the small-scale space, e.g. “go behind the desk.”

Gorniak and Roy (2004) implemented a system that kind find the referents of referring phrases given a visual scene by resolving color, spatial relations, grouping information, and anaphora. Their system, Bishop, resolved spatial prepositions with an implementation of (Regier and Carlson, 2001) to distinguish one object out of a group of distractors. The domain for this work was initially a simulated, abstracted tabletop environment of configurations of colored cones. It was later integrated with the Ripley robot in a physical tabletop environment (Roy et al., 2004). As discussed in 3.3.1, Levit and Roy (2007) implemented a system applying understanding of spatial language to accomplish the MAP-TASK.

2.4.2 Computational models of route instructions

Riesbeck (1980)’s system evaluated route instructions by high-level characteristics, independent of the environment. His natural language parsing and understanding program analyzed a set of route instructions for overall *clarity* and *cruciality* measures. Each motion

must be described completely and precisely (clarity); additional descriptions provide checks but are not crucial. The software simulated the role of a person glancing over a route instruction text, while questions can still be asked before navigating, not of an agent following route instructions in the environment.

Agre and Chapman (1990) discuss plans as communicative acts and instructions as communicating under-specified plans. They showed how route instructions do not uniquely specify action sequences, but instead constrain navigation by providing a plan skeleton, with exploration sub-goals the follower must accomplish. Chapman (1990) followed up on this theoretical paper in implementing the “Sonja” system, which interpreted spoken advice and instructions to better fight the monsters in her virtual dungeon.

Alterman et al. (1991) implemented a system which reactively replans to read the instructions when its naïve plan proves inadequate. It makes an inference graph by analyzing the keywords in instructions, simplifies the graph using graph summarization techniques, transforms the graph into a procedure, and resumes executing with the amended plan. The system, FLOABN, operated in a discrete event simulation. Example instructions focused on different ways of paying for phone calls.

Zelek (1997) implemented a system that followed spatial instructions from a small, controlled vocabulary grammar, chosen from a graphical user interface. The system was able to execute two basic commands *travel^a* and *find^a*, which are two of the actions in out current work. The system also had models for two-dimensional spatial prepositions, although how these are resolved is not detailed. The system was evaluated on a physical robot.

Webber et al. (1995) looked at the broader question of inferring an intended plan from any instructions. This work examined the linguistic and domain knowledge needed to get a virtual agent to follow instructions from various domains. They state

A plan’s relationship with a set of instructions is also not rigid. It depends, inter alia, on various features of the instructions, including: (1) whether the

instructions convey doctrine (general policy regarding behavior in some range of situations) or procedure (actions to be taken now or at some specified time in the future) ; (2) in the case of procedural instructions, whether they are given before, during, or after action; (3) whether the instructions are meant as advice, suggestion, order, request, warning, or tutorial.

Di Eugenio (1998) reports on the language system of this work. Her software analyzes general instructions, such as craft guides, matching the text against a plan library using plan recognition. The system integrates a lexical semantic ontology (Conceptual Semantics Jackendoff (1983)) and a description logic based system. The major contribution is interpreting “purpose clauses” (do *this* to accomplish *that* or do *this* such that *that* is done) (e.g. “Turn left to face the chair.”). Purpose clauses help lookup an appropriate plan in a plan database, as the purpose clause indicates the plan’s (or at least the utterance’s) goal. The system was integrated with the AnimNL system, which is a VR animation able to simulate several tasks. The self-admitted lack in the system is an inability to synthesize meaning across the discourse, instead, it interprets each sentence in its own context.

Other parts of Di Eugenio et al.’s work examined instructions for the role of free adjuncts (e.g. “Facing the chair, move forward”) (Webber and Di Eugenio, 1990), negative imperatives (e.g. “Do not go down the blue hall.”) (Vander Linden and Di Eugenio, 1996), and handling standing orders with some autonomy (Bindiganavale et al., 2000).¹

Müller et al. (2000) implemented a system that can follow a formal route description through an environment, with the intention of adding on a natural language understanding system. Descriptions follow the Tversky and Lee analysis, specifying where to turn or switch paths (Tversky and Lee, 1998).

Frank (2003) suggested formalizing verbal route instructions into action schemas, considering the “pragmatic information content” of route instruction texts the same if they

¹All examples are from our domain, not their papers.

produce equivalent actions.

A group in Bremen, Germany is building an intelligent wheelchair (Lankenau and Röfer, 2001; Mandel et al., 2005) that can share control with a human driver through a natural language interface by integrating a spatial ontology (Krieg-Brückner et al., 2004) and dialogue model into an agent control architectures (Ross et al., 2004). Other work examines how people direct a robot using natural language to one of a group of objects, particularly the dialogue strategies and spatial referencing used, and whether directors used open- or closed-loop commands (Moratz et al., 2003; Tenbrink, 2003; Tenbrink et al., 2002; Tenbrink and Moratz, 2003).

Shimizu and Haas (2006) built a system that followed instructions through a simulated building. The system parsed free-form natural language instructions into a command template of a verb of *travel^a* or *enter^a*, a landmark of a *door* or *hallway*, a direction of *left*, *right*, or *straight*, and a ordinal of which *hallway* or *door* is referenced. The routes consisted of about two segments. This work is attempting to learn to match word segments to action sequences and for the corpus gathered, succeeds at 77%. The route instructions were elicited by showing the director the route to follow and routes that the experimenter could not follow were removed, leaving a corpus which should be 100% followable.

Gorniak and Roy (2006, 2007) implemented a system that follows directives that one player gave another while solving a puzzle in a video game. Their system performs plan recognition by parsing utterances into an affordance filter, which, in turn, selects the most probable action on an object, given the utterance and situation. Placing the system in the same situations with the same linguistic inputs as a human player, the system selects the next action about 70% of the time, whereas plan recognition alone only predicts 50-60% of the next actions. This evaluation only allows the testing of one command at a time, and the commands are extremely simple, with 50% of the commands consisting of a single word.

Tellex and Roy (2006) programmed “spatial routines,” or simple procedures, to

execute single commands instantaneously to move within a room or corridor. The system understood eight commands, combinations of turn, go, and stop with parameters, such as “Go (straight|right|left)” and “go across the room.” The procedures included achieving preconditions and were evaluated by whether the simulated robot produced a similar path across the local small-scale space as people in response to the command.

2.4.3 GRACE and GEORGE in the AAI Robot Challenge

Perzanowski et al. (1998, 2001) implemented a system that combined a speech recognizer, a deep parser, a dialog model, hand gesture recognition, and a Palm Pilot control interface. A user could command the robot to move around a mapped, small-scale space by speaking and gesturing.

GRACE extended this architecture, adding the ability to follow a route instruction series through an unmapped, unknown large-scale space (Simmons et al., 2003). The robot GRACE navigated through a conference center by asking for and following route instructions. GRACE could string together several simple commands, using an instruction queue executor. They also handled implied new interim destinations (“Take the elevator”).

In 2002, GRACE successfully, though haltingly, completed the Robot Challenge at AAI 2002 (Simmons et al., 2003). The 2003 robots GRACE and GEORGE were beset by hardware, software, and communications problems that illustrate the need for more user visibility into the state of the system. Still, the robots were directed down a hallway, up a ramp and through a narrow doorway, and across an exhibition hall.

GRACE and GEORGE had several major limitations. Most debilitating, the commercial speech recognition system was unreliable. The vocabulary and sentence structure were limited so only a trained operator could direct the robots. The navigation planning code relied on having a completed global metrical map, so navigating to unseen,

unknown locations was extremely fragile. Crowds of people forming shifting walls further confused the robot.

GRACE and GEORGE did not reason to infer implied actions. They had only one interpretation of the instructions, although this was checked with the director. The robots did not estimate the likelihood of action success, but instead asked the director.

2.4.4 Instruction-Based Learning (IBL) Project

The Instruction-Based Learning (IBL) for Mobile Robots project is another implementation of route instruction following on robots . Bugmann et al. (2001) presented a corpus of 96 spoken route instruction sets from participants guiding a human operator, who had remote control of a robot navigating through a tabletop model of a town center. They modeled the instructions as action schemas, called “functional primitives,” such as MOVE FORWARD UNTIL <COND>, TURN <DIR> <LOC>, <LANDMARK> IS LOCATED <WHERE>, and GO TO <LANDMARK>. Lauria et al. (2002a) implemented a robotic system capable of following programs of functional primitives from this corpus, expanded to 144 route instructions. The 15 IBL functional primitives include procedures such as *go_until^a*, *exit_roundabout^a*, *follow_road_until^a*, and *take_road^a*, all of which would be modeled with our Travel^P procedure with various parameters.

2.4.5 Spatial Ontologies and Spatial Instructions

Software systems that analyze or follow route instructions can be distinguished by how they represent space. Freundschuh and Egenhofer (1997) survey a variety of spatial representation models and define broad categories based on (1) if the objects in the space are *manipulable*, (2) if the space requires *locomotion* to experience, and (3) the size, or *scale*, of the space. This work focuses on non-manipulable, large-scale spaces that cannot be experienced from any one perspective: the agent must turn (*panoramic space*) or move (*environmental space*) to see the space.

Other work concentrates on understanding single spatial commands in the small-scale space of a room (Skubic et al., 2004b; Tellex and Roy, 2006) or tabletop (Roy, 2005). Finally, other work follows instruction sequences in a large-scale space, but does not use spatial and linguistic knowledge to recover from instruction errors or to infer implicit actions (Bugmann et al., 2004; Simmons et al., 2003).

Chapter 3

Language and Task Corpus Methodology

The language and task corpus methodology seeks to build and evaluate systems that will be robust to natural instructions from people. *Natural instructions* are not only instructions in an unconstrained, natural language, but they also represent the kinds of instructions that people give to one another. In the real world, instructions are often under-specified, with erroneous information or some necessary actions not implicitly stated. People are remarkably robust to the challenges of natural instructions. We seek to build a system that can follow the same natural instructions that people can, especially those instructions that most people can and do follow successfully.

The language and task corpus seeks to discover which instructions the “reasonable agent” can follow by following this procedure: (1) collect natural instructions from many people, (2) give the instructions to many people to follow and rate, and (3) give software systems the same instructions to follow in the same environments. The language and task corpus can be used in development and evaluation, using human performance as a benchmark for the software system. Running systems with disabled components or alternative implementations and heuristics both measures the performance impact of each

software component and makes predictions about how people process and execute natural instructions.

The language and task corpus methodology with multiple human directors and followers is an excellent way to study human task-centered language use, as well as for system development use at the focus of this dissertation. In fact, other route instruction researchers independently suggested, but not implemented, this methodology for psychology studies of spatial route instructions. Lovelace et al. (1999) rated route instructions subjectively, but could not verify that their poorly rated route instructions were actually functionally worse. They suggested a study where subjects followed a variety of route instructions for a variety of routes in a virtual reality environment to determine this crucial question.

Our evaluation testbed ties together an instruction corpus, navigable environments, and action traces from human and artificial agents with linguistic and spatial reasoning abilities. This testbed of a route instruction text corpus tied to simulated environments presents a challenge task for researchers in natural language understanding and spatial reasoning. The methodology emphasizes understanding the gist of route instructions over some details: the essential linguistic and spatial details separate navigation success from failure. However, to be tested, components must be integrated into a complete agent that can read the instructions and apply the understanding to act in the world.

This paper contributes an assessment of human performance for communicating route information through unfamiliar large-scale spaces. By comparing the performance of a computational model with and without the ability to infer implicit procedures, we measure how often understanding the unstated is necessary to succeed in this task. Though this ratio will change for other tasks and domains, the methodology of comparing human and automated systems on corpora of problems will generalize.

We believe that the language and task corpus methodology described here will generalize to instructions about other complex procedural tasks, including cooking, first aid,

furniture assembly, automobile repair, and many others. We believe these tasks should be similarly evaluated, with a testbed that demonstrates sufficient understanding by achieving a complex, situated task given diverse natural language instructions.

3.1 Language and Task Corpus Methodology for Spatial Route Instructions

Several decisions were made in how to gather the instruction corpus to elicit more natural instructions. Each of these factors increases the ecological validity of the instruction corpus (Cohen, 1995; Rosenthal and Rosnow, 1991).

Directors learned the environment through a combination of undirected, free exploration and directed navigation task execution. In the free exploration phase, the directors could move around the environment in any pattern, discovering named positions by moving into them. The subjects had to actively chose how to move through the environment, exploring through a first-person perspective. Subjects neither saw a map, nor were guided in their exploration.

In some other studies, subject passively observe or are led through a recorded route through the environment (Lovelace et al., 1999; Shimizu and Haas, 2006). Cognitive psychology studies have shown that active navigation leads to qualitatively different and better learning of the environment than passive observation of navigation, the “passenger effect” (Dayan and Thomas, 1995; van Asselen et al., 2006).

In the navigation quiz phase, directors were placed at one of the named locations and asked to navigate to one of the other named positions. This both helped the subjects find parts of the environment they may not have found. More importantly, it forces subjects to reason and problem solve using their cognitive map. Subjects must demonstrate competence in way-finding among the named positions throughout the environment. During the trials, the experiment program automatically provides feedback after each trial

on whether the subject finished at the destination or if the route navigated was circuitous. However, subjects must learn how to overcome these deficiencies on their own.

In the last phase, the directors give instructions. Directors are placed at the starting location and allowed to turn so that they can recognize the start. Then the directors must plan and describe a route to another position somewhere in the environment. While writing the instructions, the environment is not visible. The directors are not shown the route to describe, but must apply their knowledge of the environment to plan the route. After giving the instructions, the directors are asked to follow them and rate themselves. With this information, we can measure when a director unknowingly described a route to a destination other than the one requested.

The directors provide the instructions as typed natural language text, rather than as a spoken monologue or a spoken or written dialogue. Writing reduces the disfluencies in the instructions by allowing the subjects time to think and edit. Since the instructions are given to unknown followers, directors cannot establish *ad hoc* conventions with the followers (Anderson et al., 1991). The director must give general instructions to a follower without knowledge of the environment. The problem of using dialogue to find or establish common ground in mental representations is interesting, but is a substantial task in itself (Anderson et al., 1991; Cohen, 1984; Edmonds, 1993, 1994; Garrod and Anderson, 1987; Garrod and Doherty, 1994; Heeman and Hirst, 1992; Pickering and Garrod, 2004; Schober, 1993).

To understand how people describe routes in large-scale environments, we performed a series of experiments in virtual environments. The use of virtual environments had several benefits. First, all participants were guaranteed to have no initial experience in these environments. Second, we could record the learning process and exploration patterns of the directors. Third, we could move followers between environments easily, quickly, and without notice, to discourage learning the environments. Fourth, we could directly compare the performance of people navigating these virtual environments against an artificial agent navigating the same maps given the same instructions.

The routes range from one travel action to many turns through complex environments. The followers had to identify the destination only from the route instructions, not from any distinctive marking or the trial automatically ending. Moreover, the instructions were far from perfect with some providing minimal guidance and many with significant errors in turn direction, object identity, or distance estimates (MacMahon, 2005).

3.2 Related methodologies for evaluating natural language understanding systems

As we discussed in 2.4.4, Bugmann et al. (2004, 2001) enacted a similar methodology for their Instruction-Based Learning project. The work in this paper is more easily and less expensively replicated, since no special robotic equipment or physical town model is needed. More importantly, our subjects learned the environments from the same first-person perspective as the human and software agents following the instructions and wrote instructions from memory. Bugmann’s participants only saw an outside, panoramic perspective of the town model while directing.

This difference in how environments are learned and perceived between the directors and followers leads to a class of errors not present in our approach. Specifically, directors may refer to information unavailable to followers. Conversely, while our directors may make errors while learning the map through navigation or recalling the map while directing, these errors are cognitively interesting and prevalent in the real world. Previous work has found differences in the types and rates of errors that directors make when the director is looking at a map – “map-present condition” – or directors describing the route from memory – “map-not-present condition” (Brown et al., 1998; Ward et al., 1986).

In “Wizard of Oz” studies, another common paradigm, the experimenter simulates user interaction with a software system by having an expert “man behind the curtain” control the behavior of the system. The Instruction-Based Learning corpus was gathered

through a “Wizard of Oz” methodology, as were several other prominent corpora of spatial language interaction with software systems (Green et al., 2006; Perzanowski et al., 2003; Skantze, 2005). The problem of “Wizard of Oz” studies for instructions is that the user interacts with one expert user, who may not have typical reactions to the instructions. Worse, it is just one interpretation of the instructions, not covering the individual differences in language interpretation or navigation strategy. A variant of the “Wizard of Oz” method uses a static scripted script for the system interaction (Tenbrink, 2003).

A similar strategy is to have live interaction with a live system under development (Fischer, 2003; Hüttenrauch et al., 2004; Moratz et al., 2003; Tenbrink, 2003; Tenbrink and Moratz, 2003). This has the same problems of the “Wizard of Oz” studies compounded by the fact that the follower is now a partially developed software system. In fact, these studies find that directors can spend a lot of effort trying ascertain what linguistic and spatial abilities the system has. The interactions are not natural, but qualitatively differ from interactions with people, as those who run these studies have found. When the system is fairly developed, this can be a good evaluation, but it does not work well for development.

3.3 Related methodologies for studying natural language tasks

One common method of studying natural language tasks is for experts to annotate the corpus to create a treebank (Ellsworth et al., 2004; Johnson and Fillmore, 2000; Kingsbury et al., 2002; Palmer et al., 2005; Vander Linden and Di Eugenio, 1996). This methodology grows from the linguistic tradition and is thus best suited to study the language aspects of the tasks, but less so for the action portions.

In our study, the directors learn the environment from a first-person view of self-directed free exploration. The directors then describe the environment from memory and the followers navigate using the instructions with the same perspective the directors had. In contrast, in other experiments the directors learned the environment from an overhead view

or map, often observed while giving instructions (Bugmann, 2003; Bugmann et al., 2004, 2001; Kyriacou et al., 2002, 2004; Lauria et al., 2001, 2002a,b).

Other route instruction studies (Lovelace et al., 1999; Taylor and Tversky, 1992; Vanetti and Allen, 1988) rate route instructions subjectively, but do not test if navigation success is affected. In these studies, the experimenters code expected errors, but do not measure when people are affected by those errors when following the instructions.

In other experiments, subjects are guided through a route then told to describe it (Lovelace et al., 1999; Shimizu and Haas, 2006) or rely on previous experience outside of the experiment to give the participant familiarity with the route(s) (Daniel et al., 2003; Denis et al., 1999; Lovelace et al., 1999; Michon and Denis, 2001; Nothegger et al., 2004). Evidence has shown that different mechanisms are used in learning an environment or a route from maps versus from first-person experience (Garden et al., 2002), as well as much research into how people pick the routes to describe (Dalton, 2003; Duckham and Kulik, 2003; Haigh et al., 1997; McDermott and Davis, 1984).

The environments are virtual large-scale indoor environments. From any place in the environment, only a limited portion of the full environment can be seen. The environments have a maze-like layout, consisting entirely of corridors and intersections, with no large open areas. Other work has focused on the problems of understanding single spatial commands in small-scale spaces, such as navigating within a room or open field (Blisard and Skubic, 2005; Blisard et al., 2006; Skubic et al., 2004a,b; Tellex and Roy, 2006) or in manipulable space, such as a desktop (Pook and Ballard, 1996; Roy, 2005; Winograd, 1972; Yu and Ballard, 2004).

3.3.1 MAP-TASK Corpus Methodology

A classic psycho-linguistic experiment produced the MAP-TASK corpus (Anderson et al., 1991). The MAP-TASK is similar to following route instructions through large-scale spaces to some degree: participants use spatial language to describe and recreate a route. However,

looking at the details, this is a very different task. The participants in the MAP-TASK use the language of large-scale spatial navigation instructions — and some explicit references to the paper medium — to describe the task of drawing a line on a map. The follower in the MAP-TASK is performing a task akin to a would-be tourist at home, reading a Paris guide book and tracing a route on a map; in the route instruction following task, that tourist uses that book to navigate the streets of Paris.

The MAP-TASK does not involve navigation, as both participants interact with the world only through a map. In the MAP-TASK, the participants have equal *a priori*, though differing, knowledge of the environment. Each has a map containing slightly different overhead sketches of the environment. Each can see their entire map at all times and the only way to gain additional information about the environment is by talking to the other participant, not through action in the environment. The landmarks are sketched onto the map and labeled with descriptive noun phrases, e.g. “vast meadow,” so there is no variability in the noun phrase used to name referents or the landmarks referenced. The director instructs the follower to recreate a route drawn only on the director’s map on the follower’s map. The route is provided by the experimenters, not planned by the director, and is visible to the director throughout the experiment.

In the MAP-TASK, the participants have equal knowledge of the environment, which both subjects view as a image annotated with labels, and they collaborate in real time to duplicate the route line shown to one participant. In the MAP-TASK, the evaluation of success is unclear: is the task is to exactly or qualitatively reproduce the route line? How should the evaluator account for the missing landmarks on the Follower’s map?

In our task, one participant has learned the environment through navigation, and must plan a route from memory, then write a text describing the whole route to an unknown follower, who will later use instructions to navigate to the destination. The director chooses what to describe and how to describe it without guidance from the experimenters like the supplied labels in the MAP-TASK.

The advantages of the MAP-TASK are that it controls for some of the individual differences in experience and ability on some dimensions of the task. The directors' and followers' only knowledge of the environment is what they see on the map and what they hear. Thus, the experimenters control both participants' initial state of knowledge. Additionally, by selecting and labeling the landmarks, the experimenters greatly reduce the lexicon and referring phrase variation. Finally the MAP-TASK uses paper (or any other image display) as its only apparatus, making it easy to replicate.

Levit and Roy (2007) created a system to follow the instructions in the MAP-TASK corpus. Their system uses a dynamic programming approach to combine manually modeled and grouped "Navigation Information Units" to draw the best-fit graphical path on the director's map. The system relies on the fact that it has, *a priori*, perfect and complete information from the graphical map, which was exactly the information the director had. The evaluation does not compare to the paths other people drew given the instructions, only the experimenter provided reference path. The system was also not evaluated drawing paths on the follower's map, unlike the human followers in the MAP-TASK experiments.

Chapter 4

Human Route Instruction Experiments

Instructions imply two roles: the *director* and the *follower*. The director plans to accomplish the task and describes the procedure in the instructions. The follower understands the instructions and executes the procedure in the environment to accomplish the task.

Instructions from multiple directors will vary in errors, omissions, vocabulary, grammar, and information provided. When the environment allows multiple solutions to a task, the procedures described will also vary. Additionally, the directors may have learned different aspects of the environment, and have different strategies to describe tasks in the environment. One goal of this study is collect a corpus covering all of this variability.

Human followers are intelligent and actively attempt to match the directors' intent to the task and environmental context. Followers also will have individual variations in how they interpret and execute instructions. The follower can often achieve the task despite gaps and explicit errors in the instructions. The crucial question for instructions is how often followers are able to accomplish the task. By giving many followers instructions from many directors over multiple tasks, we can measure how often people reach the destination across a wide variety of instructions.

4.1 Motivating questions

As detailed in Chapter 2, research in several fields has examined giving and following route instructions. Some researchers gather a corpus of instructions and have experts rate the instructions and code them for various attributes, e.g. landmark usage and errors in description. Other studies take a small number of instructions and have people navigate using the instructions through an environment. No other study gathers a corpus of instructions over many routes from one set of people and then have another set of people follow and rate the instructions. We examine the behavior of directors and followers in conjunction, but independently.

We want to measure human variability in instruction-giving and instruction-following for complex routes through an unknown, large-scale space. This corpus allows us to answer the following questions: How do directors vary in describing the routes and how do followers vary in following the routes? How do subjective ratings correspond to objective success rates in following the instructions? What is the distribution of good and poor instructions? Does a link between gender and spatial route instruction performance exist?

To measure human linguistic and spatial behavior when giving and following spatial route instructions, we gathered a language and task corpus from human subjects in three experiments. The corpus consists of instructions from multiple directors over multiple routes that are followed and rated by multiple human followers.

4.2 Study overview and motivations

All three of our human studies share the same basic procedure. Directors learned the environments and wrote instructions. Followers read these instructions and followed them through the environments. Each of these roles is summarized here and described in detail later in the chapter.

Directors learned a virtual environment through unguided, first-person exploration. Each director was tested by navigation tasks, to ensure he or she had sufficiently learned to navigate through the environment efficiently. Finally, each director planned the route from memory, typed route instructions, followed the route themselves, and rated their own instructions for each of up to 42 routes through the environment between two named positions.

These instructions were later followed and rated by a separate set of human followers. The followers did not have prior experience in these environments. Each follower followed instructions from multiple directors in all three environments, without ever repeating exactly the same route.

In Experiment 1, we investigated how instructions vary across a small number of directors and whether a director's style and quality varies across environments. Six directors each learned and gave 42 instructions in each of three virtual environments, a total of 126 sets of instructions each, (756 requested instructions across all 6 directors). Each set of instructions was followed and rated by six other people.

In Experiment 2, we wanted to measure how much of human variability our six directors covered. In Experiment 2, we had twelve directors each learn one environment and asked for 42 routes from each. Multiple people followed and evaluated each set of instructions from Experiment 2, as well as following and evaluating a sampling of instructions from Experiment 1.

In Experiment 3, we tested how subjects' controlled movement affected instruction-following performance. In the previous studies, directors and followers moved through the environments using discrete actions triggered by the keyboard. For instance, pressing '8' moved the camera along the corridor at a constant speed to the next possible intersection, unless a wall was ahead. In Experiment 3, both the directors and the followers moved the camera using continuous joystick control. In Experiments 1 and 2, the camera moved smoothly down the center of the hallways, but in Experiment 3, each person controlled the

speed and heading continuously. This question addresses how the discrete control affected the route instructions.

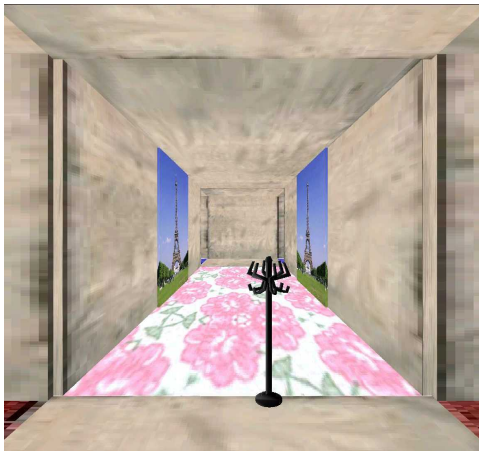
4.3 Apparatus

These experiments are performed in the Vizard 2.53g virtual reality engine (Vizard, 2006). Vizard provides an immersive environment with optical flow when moving, photo-realistic textures, and three-dimensional objects. The experiments are run on Dell desktop computer with a 17" monitor under 1248x1024 resolution. Directors heard audio cues from either desktop-mounted speakers or headphones. Followers had no audio cues.

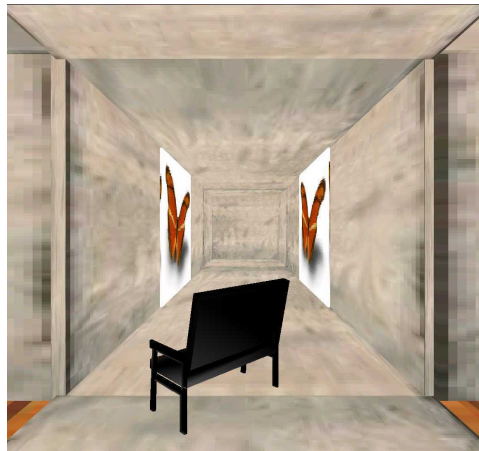
4.3.1 Control of movement through the environment

In Experiments 1 and 2, the numeric keypad controlled motion between places and poses at a place. If the participant pressed the '8', the view moved forward down a hallway to the next place, or remained in place if a wall was immediately in front. The '4' and '6' keys rotated the view 90° to the left and right, respectively. The views were placed so that the participant could see the presence or absence of any hallways immediately to their right and left. See Figures 4.2 and 4.1 for example views.

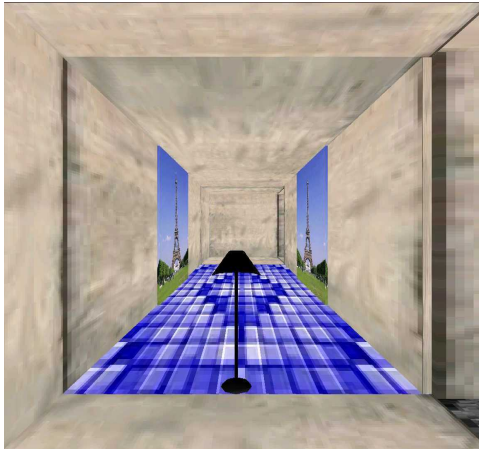
In Experiment 3, the participants used a joystick to navigate through the environment, controlling the speed and direction by direction and angle of the joystick. Unlike the discrete condition, movement was controlled in an analog fashion, where the further the joystick was angled, the faster the camera moved through the virtual space. Additionally, the participant could turn to any angle and could stop motion or turn at any location. In the discrete motion condition, motion only stopped at designated poses.



Hatrack



Sofa



Lamp



Chair



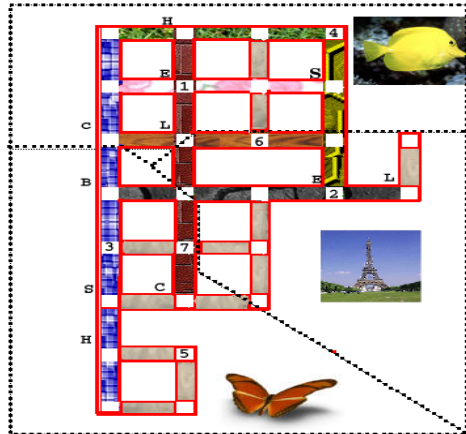
Easel



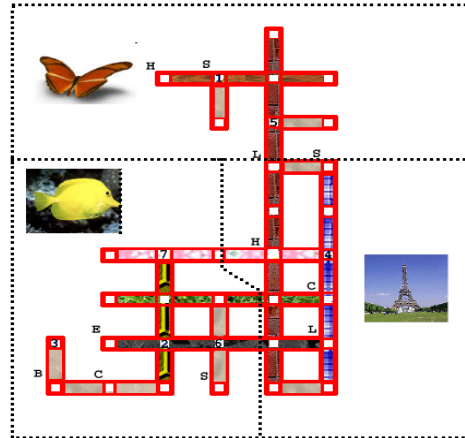
Barstool

Figure 4.1: Sample views of all objects, from the *Medium* environment.

Compact



Medium



Sparse

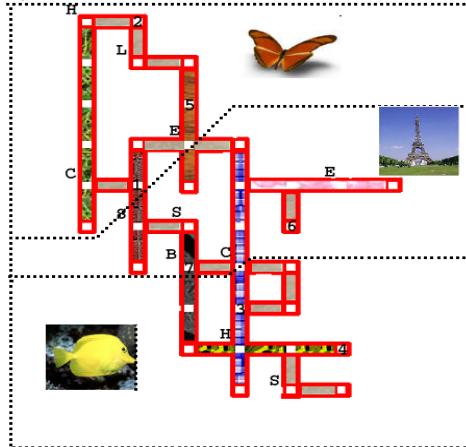


Figure 4.4.1: Maps of the three virtual environments, which participants explored from a first person perspective. Participants did not see these maps or any global representation of the environments. The three regions marked by dotted lines each have a unique wall hanging: fish, butterfly, or Eiffel Tower. Each long hallway has a unique flooring. Letters in the maps mark objects (e.g. ‘C’ is a chair). Numbers indicate the named positions.

4.4 Stimuli

4.4.1 Environment Maps

These environments and the experiment control software build on top of previous studies on spatial navigation (Kuipers et al., 2003; Stankiewicz and Eastman, 2008; Stankiewicz and Kalia, 2007; Stankiewicz et al., 2006, 2001). The environments were generated on a Cartesian grid and modeled in Virtual Reality Modeling Language (VRML).

The tests were run in three distinct environments, which were constructed from the same components. Figure 4.4.1 shows the global layout of each environment. All of the

<i>Name</i>	Number of positions (Dead, Mid, Int)	Number of straight paths	Average path length: Mean (Median)	Percentage unforced decisions	Mean ints per path length
<i>Compact</i>	28 (1, 0, 27)	15	2.7 (2)	71%	0.97
<i>Medium</i>	34 (10, 4, 20)	14	2.9 (2.5)	44%	0.72
<i>Sparse</i>	37 (8, 7, 22)	19	2.1 (1)	30%	0.81

Table 4.1: Characteristics of the three testing environments. Positions are either *Dead*-ends, *Mid*-path, or *Intersections*. Paths are defined by continuous straight segments.

participants saw the environments from the first-person perspective of the virtual reality rendering. The three environments varied in the density of the layout, as measured by the shortest travel routes between the named positions and average hallway length. We name each environment according to its density. The most compact (*Compact*) had a mean shortest route length of 4.2 (median 4), the most spread-out (*Sparse*) environment, mean 6.0 (median 6). The shortest route was a minimum of one travel action, the longest was a minimum of 13.

Table 4.1 shows more of the statistics of the environments. Each environment has paths totaling 40 segments. The environments contain between 28 and 37 positions. The environments have between 14 and 19 paths, where a path is defined as hallway segments connected along a line.

Certain positions in the environments were marked with audio cues for the directors, the *named positions*. When the director encountered a named position in the Exploration phase, and at the beginning of the Navigation Quiz and Route Instruction Entry phases, a voice announced the position name, e.g. “*Position 2*” or “*Position 4. Go to Position 7.*” These positions are marked by the numbers on the maps in Figure 4.4.1. There were two sets of named positions. In Experiment 1, all directors learned and described routes from Position Set 1. In Experiments 2 and 3, half of the directors learned Position Set 1 and half Position Set 2.

The layouts vary considerably. *Compact* is densely connected with few short or dead end paths. *Compact* has 96% of possible intersections connecting at least two paths.

This count of intersections excludes positions that are only on one path: dead-end positions and positions in the middle of a hallway where the participant cannot turn onto another path. *Sparse* has an intermediate degree of connection, with 59% of possible intersections connected and many short paths and a long, sparsely connected loop. *Medium* has also has 59% of possible intersections connected, with many dead ends off of relatively long paths. Finally, in the *Compact* environment, nearly three-fourths of the positions are decision points, where a navigator has more than one way to proceed forward, while less than one-third of the positions in *Sparse* present the way-finder with a decision.¹

4.4.2 Environment Landmarks

To provide useful cues for the directors, we placed 11 three-dimensional objects of 6 different types in each environment. The types of objects were an easel, a hatrack, a sofa, a chair, a stool, and a lamp. See Figure 4.1 for sample views of the different objects. The objects were placed at potential intersections in the environments. Some objects were repeated within an environment and all objects occur in each environment. For instance, Figure 4.2 shows two different easels in the same (*Compact*) environment. The positions of the objects are marked by letters on the maps in Figure 4.4.1.

The objects were chosen to be easily identified, common objects that could be shown at an apparent normal scale in relation to the hallway size and view height. They are all indoor furniture. Several are from a similar category of human seats: the chair, barstool, and sofa. The lamp and hatrack appear similar from a distance, but are distinct on a close view. The easel is more semantically and perceptually separate from the others, but occurs less often in everyday life than the other objects.

Furthermore, each environment was divided into three separate regions, designated by distinct pictures on the walls (see Figure 4.4.1). Finally, 7 long hallways within each environment had a visually distinct texture mapped onto the floor. Figure 4.4.1 shows the

¹Formally, a place with more than two gateways (Kuipers et al., 2004), eliminating dead-ends and corners.

layout for the three environments. An immediate forward view of each texture hallway can be seen in Figure 4.2.

The hallway flooring textures were chosen to be distinct in both color and pattern. All textures were chosen to be recognizable floor or ground patterns. The non-distinct cement texture is the only one repeated on separate path segment floors within an environment, and is also used for walls and ceilings. The cement texture was used on each path that was one or two segments long.

4.5 Human Directors Learn, Navigate, and Describe

All three experiments share the same procedure for learning an environment and writing instructions for routes through it. We will describe the common director procedure and then describe the participants and manipulated variables of the individual experiments.

4.5.1 Procedure

The directors progressed through four phases, described in detail below. First, in the *Introduction* phase, the directors were briefed on the experiment, answered some demographic questions, and were acclimatized to the virtual environments, navigation interface, trial procedure, and text entry. Second, in the *Exploration* phase, the directors explored the environments by moving through the environment for a fixed distance traveled, equal to 120 hallway segments. Third, in the *Navigation Quiz*, the directors were asked to navigate between pairs drawn from the seven named positions that were announced in the exploration phase.

If the directors demonstrated that they could reach each position and navigate relatively efficiently (within 65% of the shortest distance for that route), they progressed to the *Route Instruction Entry* phase. Otherwise, they repeated the *Exploration* and *Navigation Quiz* phases.

In the *Route Instruction Entry* phase, the directors were repeatedly placed at a

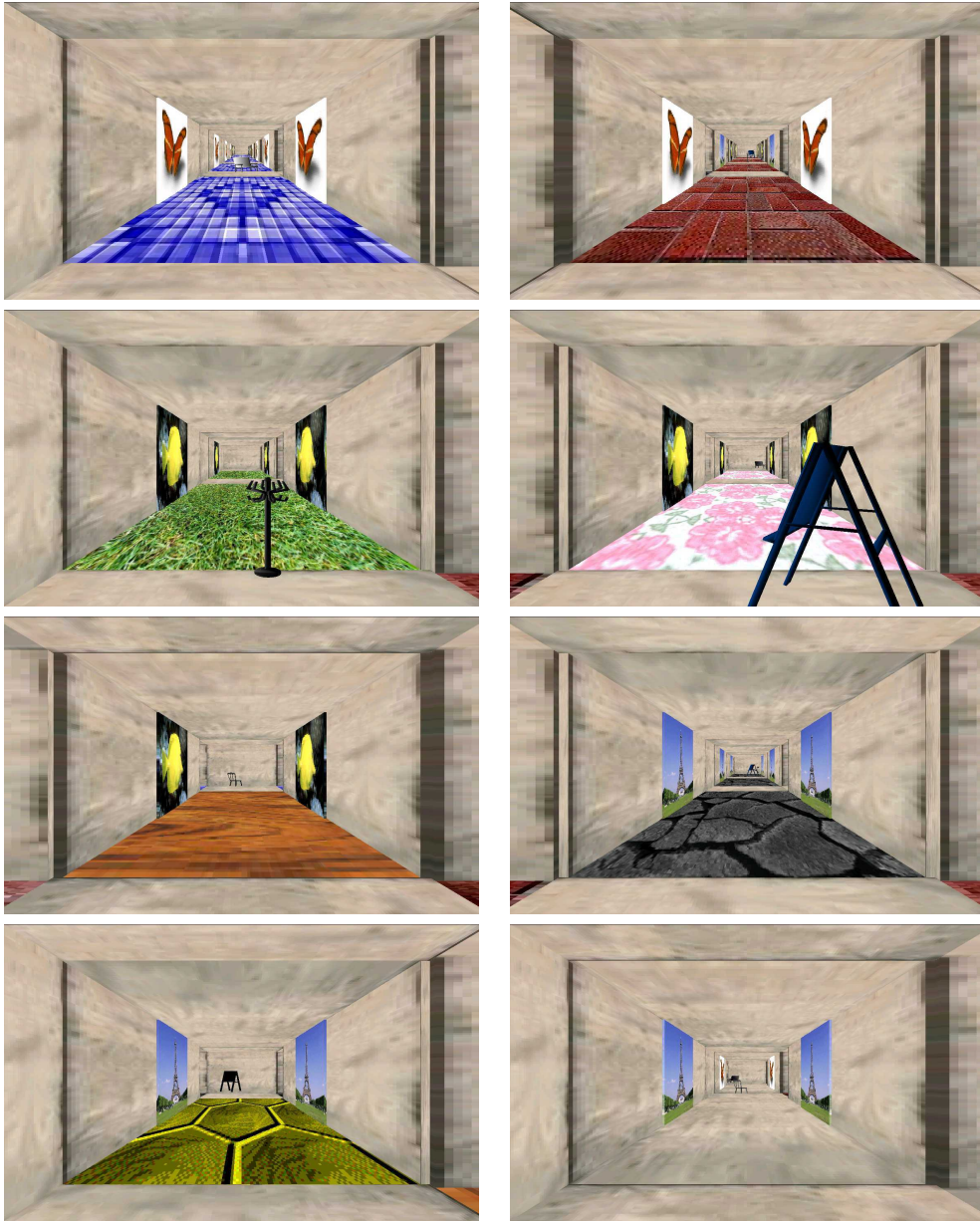


Figure 4.2: Sample views of all hallway textures, from the *Compact* environment.

named position and asked to type instructions to another named position into a text entry window (See Figure 4.3). After entering the instructions, the director navigated the route he or she just described and rated their own instructions (See Figures 4.4 and 4.5).

Introduction Phase

First, each participant was briefed on the design and goals of the experiment and were asked for informed consent. See Appendix A for this consent form.

Second, the participant was asked a number of questions, including

- Is your hometown rural, suburban or urban?
- Are your hometown's streets not, mostly, or entirely laid out on a regular grid?
- How much first-person gaming experience do you have?
- Do you get motion sickness?
- What is the language you speak at home?

Next, the participant read an instruction sheet explaining how the experiment will be run. See Appendix A.4 for the written instructions they received.

Finally, the participant was taken into the virtual reality lab and placed at a desktop computer. The participant was placed in a small demonstration environment and familiarized with the movement and audio cues. The participant ran through an abbreviated version of all three phases of the study in the demonstration environment: Exploration, Navigation Quiz, and Route Instruction Entry. Once the participant was comfortable with the experiment apparatus and procedure, the participant started the experiment in one of the three test environments.

Exploration Phase

In the *Exploration* phase, the directors learned the environment through free exploration. During this – and only this – phase, audio cues announced each of seven named positions whenever the director crossed one while moving through the environment. Until the director entered the immediate vicinity of a named position, there was no indication that the place was a named position.

During this phase, the director was placed at one of the seven named positions in the environment and moved through the environment without guidance. After moving a distance equivalent to 120 hallway segments (the distance between possible intersections), the director's knowledge of the environment was tested in the Navigation Quiz.

Navigation Quiz

Before asking for instructions, we wanted to ensure the directors had learned to navigate around the environment adequately. The Navigation Quiz tested the director's ability to efficiently way-find in this environment on the routes between the named positions. In the Route Instruction Entry phase, the directors will need to plan these routes from memory and describe them in instructions.

In the *Navigation Quiz*, participants were placed at a named position, facing a random direction. The position name was announced by the computer. Participants were told to turn around to ensure that they recognized the starting location. Once the participants had orientated themselves, they pressed the '0' key. At this point, the participant was asked to navigate to one of the other named positions. During this navigation, the audio cues for the named positions were off. Once the participants believed that they had reached the destination, or had given up, they pressed the space bar.²

²Note that this differs from earlier spatial navigation experiments using this software (Kuipers et al., 2003; Stankiewicz and Eastman, 2008; Stankiewicz and Kalia, 2007; Stankiewicz et al., 2006, 2001), where the trial ended automatically when the participant encountered the destination, without the participant making any explicit termination action.

The color of the ‘curtain’ that covered the screen between trials was color-coded to indicate success. If a director did not end at the correct target, the screen turned red. If the director terminated at the intended destination, but took an inefficient path, the screen turned blue. Finally, if the route was fully satisfactory – relatively efficient and correctly terminated – the screen turned green.

Navigation efficiency was measured by dividing the shortest path travel distance between two positions by the forward distance that the traveler actually moved. Thus, navigation efficiency normalizes the distance traveled by the length of the route. A traveler was deemed to have competently navigated the route if the navigation efficiency is 65% or higher for that route.

After the directors were quizzed for one route, they pressed the space bar to move on to the next trial. To pass the Navigation Quiz, the directors had to correctly navigate to each named position at least once, with average navigation efficiency above 65%. If either (1) the director misidentified any destination four times or (2) at the end of 25 trials, the director had more inefficient than efficient routes, the director participated in another Exploration phase followed by another Navigation Quiz.

4.5.2 Route Instruction Entry

During the *Route Instruction Entry* Phase, directors typed instructions, followed the route, and rated their own instructions on quality and confidence of reaching the destination. Since there were 7 named positions in each environment, there were 42 possible routes in each environment ($\binom{7}{2}$).

The directors began each trial with a blank screen and pressed the space bar when ready, which revealed the environment. The director was placed at the start of a route and given an audio cue that announced the current position name. After this, the position name announcements are turned off for the remainder of the trial. Directors were instructed by another audio cue to turn around, to orient themselves. The directors was allowed to turn,

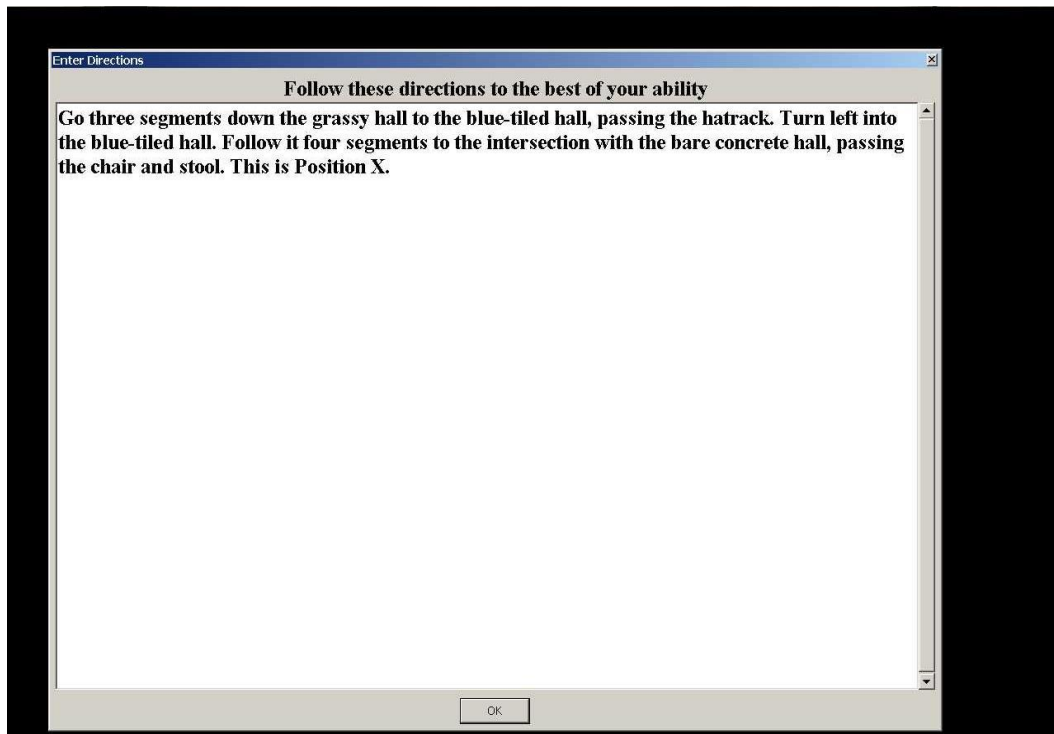


Figure 4.3: Example of route instruction window, with entered text.

but not move forward. When they were oriented, they pressed the '0' key or space bar.

Next, the screen went blank and the directors were asked to type instructions to guide another subject from the current position to one of the other named positions in the environment. The top of the window said "Enter directions to get from Position X to Position Y," where 'X' and 'Y' are the names of the starting (current) and destination positions. The directors typed into a text editor window, which allowed typing and editing text, including moving the cursor. The text entry was unconstrained, free form text, including newlines. See Figure 4.3 for an example of the text entry window, as presented in the follower part of the study.

The director was instructed that each set of instructions had to stand alone, because they would be followed by others in an arbitrary order and mixed with the instructions from other directors. The follower would be familiar with these kinds of environments, but not

this particular layout. The follower would be placed at the starting position, but facing an arbitrary direction.

After the directors typed their instructions, they clicked the 'OK' button. At this point, the environment was revealed and the director was asked to navigate from this position to the specified destination. At the end of navigation, the directors pressed the space bar to indicate they were finished.

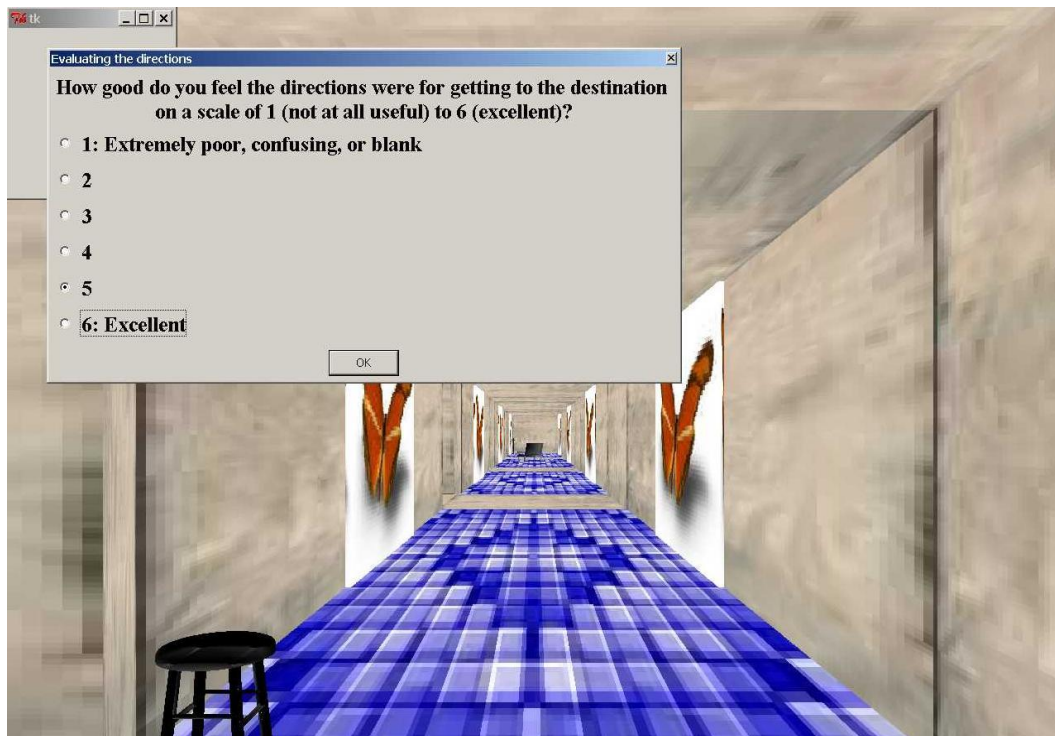


Figure 4.4: Dialog box for rating route instructions.

After navigating, directors were asked two questions (Figures 4.4 and 4.5): (1) How certain are you that you've reached the target position? (2) How good do you think your instructions were? Each question was answered on a Likert scale from 1 to 6. A scale with six discrete points was chosen so that the participants were forced to rate the instructions towards the good or bad side, with no absolutely neutral rating, as would be possible with an odd number of rating possibilities.

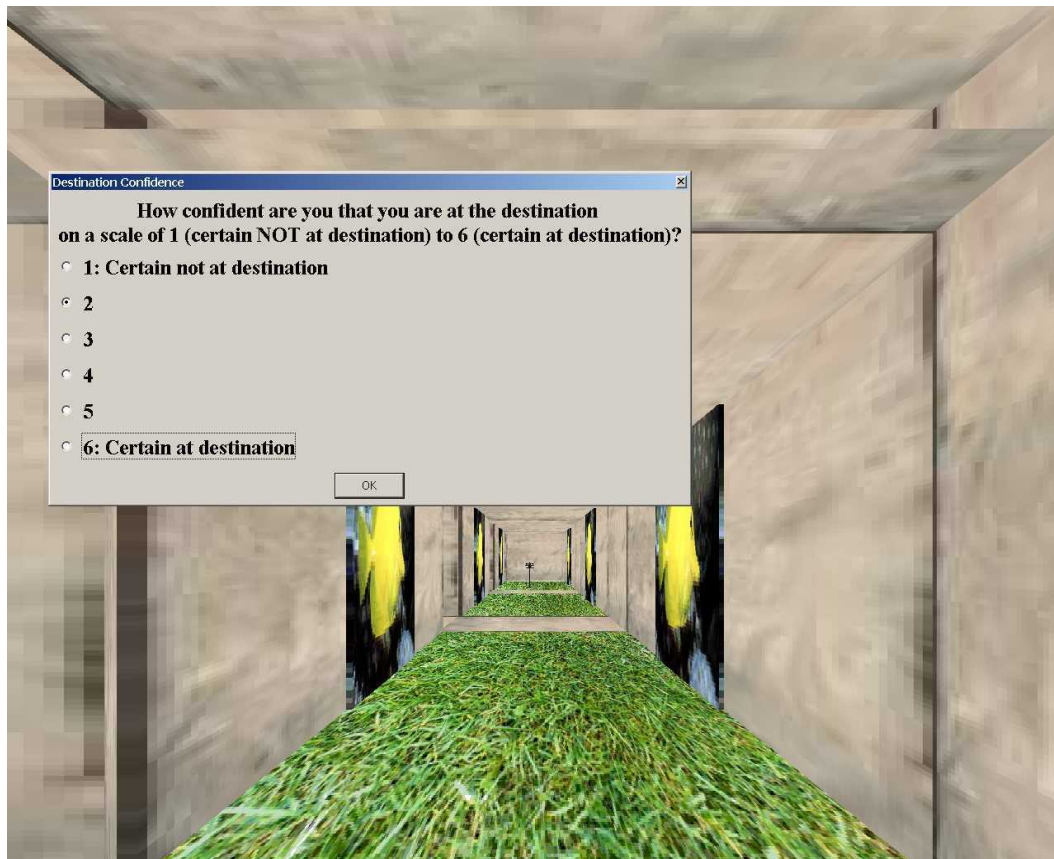


Figure 4.5: Dialog box for rating navigation confidence.

After self-rating their own performance on these two dimensions, the screen was blanked again and the director moved for the next trial, if any. The director was allowed to take any breaks whenever necessary while the screen was blank between trials. When the directors gave directions for all routes in the environment, the experiment announced “Finished!” and exited.

4.5.3 Experiment 1: Six directors across all three environments

Experiment design

Experiment 1 is a within-subject design. Each director performed three sessions, one for each environment, on separate days. In each session, the director performed the Exploration, Navigation Quiz, and Route Instruction Entry phases, including multiple Exploration and Navigation Quizzes, if necessary. The six directors were each asked for 126 route instructions over all three sessions, 42 in each of the three environments, for a total of 756 possible instructions.

Participants

The six participants were students at the University of Texas at Austin or college-graduate community members. They were paid for their participation, which took six to twelve hours. The participants range in age from 21 to 29 (mean 24.0; median 22.5). By design, there were equal numbers of males and females, three each.

4.5.4 Experiment 2: Twelve directors each in one environment

Experiment design

The experiment is a 3 (*environments*) x 2 (*position sets*) x 2 (*genders*) design, for a total of 12 directors. The combination of conditions is in Table A.1.

Participants

The twelve participants were drawn from students in the introductory psychology class at the University of Texas at Austin. They received course credit for their participation, which took about two hours. The participants range in age from 18 years, 2 months to 26 years, 2 months (mean 19.7; median 19.2). By design, there were equal numbers of males and females, six each.

4.5.5 Experiment 3: Twelve directors with continuous motion control

There are several possible differences between the discrete and continuous motion. With discrete motion, all movement was controlled and homogeneous. All turns took the same time to cause the same change in angle; likewise all travels went the same distance at the same speed. With continuous motion, the participants could travel and turn at different speeds. They could stop turning or traveling at arbitrary poses.

With discrete motion, some participants used the discrete movements as a count of distance, e.g. “Go forward three clicks.” With continuous motion, the actions were not inherently quantized or countable, though directors may still count intersections and turns or estimate travel time.

Experiment design

The experiment is a 3 (*environments*) x 2 (*position sets*) x 2 (*genders*) design, for a total of 12 directors. The combination of conditions is in Table A.1.

Participants

The twelve participants were students from the introductory psychology class at the University of Texas at Austin. They received course credit for their participation, which took about two hours. The participants range in age from 18 years, 8 months to 24 years (mean 20.0; median 19.4). By design, there were equal numbers of males and females, six each.

4.6 Route Instruction Corpus Language Statistics

For some routes, the director either did not enter any text or only entered a comment, e.g. “I don’t know.” We omit training routes, empty route descriptions, and instructions where

Name	Instructions	Vocabulary	Unique Voc.	Mean Words	Mean Sentences
<i>Corpus 1</i>					
EDA	126	163	15	27	5.7
KXP	75	182	40	20	3.2
WLH	124	181	13	41	5.8
EMW	124	280	60	63	7.9
KLS	122	176	17	45	4.2
TJS	120	190	19	20	2.2
<i>Corpus 2</i>					
JJL	30	124	13	55	5.2
JXF	42	153	11	33	3.5
MXM	24	98	4	33	3.6
MJB	42	187	19	66	5.6
PXL	41	186	25	40	2.9
QNL	42	114	7	22	3.2
BKW	42	148	9	40	3.9
BLO	38	90	6	27	3.4
JNN	24	174	19	64	6.0
LEN	42	141	8	39	2.7
MXP	42	116	6	34	2.9
TXG	34	144	20	47	4.4
<i>Corpus 3</i>					
JTM	20	148	11	59	5.0
KAJ	41	120	9	44	4.1
KXK	42	109	5	33	2.6
MHH	37	119	4	28	3.6
RRE	18	83	2	27	2.9
WAB	41	119	2	28	2.5
ARL	42	166	11	48	3.3
JLM	40	158	20	42	4.0
JXL	42	108	5	26	2.9
LCT	32	180	22	35	3.4
SCD	33	165	13	63	7.5
SMA	42	140	7	31	1.9

Table 4.2: Statistics per group of instructions by each director.

Group	n	Words	Vocabulary	Sentences	Instructions
Corpus 1	6	36.5, SEM:6.9	195.3, SEM:17.3	4.8, SEM: 0.8	115.2, SEM:8.1
Corpus 2	12	42.1, SEM:4.0	139.6, SEM:9.3	3.9, SEM: 0.3	36.9, SEM:2.1
Corpus 3	12	39.3, SEM:3.7	134.6, SEM:8.5	3.7, SEM: 0.4	35.8, SEM:2.5
Discrete	18	40.2, SEM:3.5	158.2, SEM:10.4	4.2, SEM: 0.4	63.0, SEM:9.4
Male	15	37.7, SEM:3.6	139.1, SEM:8.9	4.0, SEM: 0.3	49.7, SEM:8.6
Female	15	42.0, SEM:3.6	158.4, SEM:11.3	4.0, SEM: 0.5	54.6, SEM:9.1

Table 4.3: Corpora text statistics averaged per director group, comparing instructions per experiment and comparing instructions from male and female directors across all experiments. SEM is the Standard Error of the Means.

a director had described the same route previously.³

Tables 4.2, 4.3, 4.4, and 4.5 show the summary statistics for the instructions. For each director, Table 4.2 shows the total number of instructions written, the vocabulary as the count of distinct tokens used, how many words are unique to this director, the mean number of words used by the director (total words divided by total instructions), and the mean number of sentences, after splitting run-on sentences (as explained below). Note the large differences in length and vocabulary across directors.

Table 4.3 shows the mean numbers of words, vocabulary used, sentences, and instructions for several groups of instructions. ‘Words’ is the mean of the mean number of words used across all instructions per director. ‘Vocabulary’ is the mean number of distinct tokens used, including misspellings, but not punctuation. ‘Sentences’ is the mean number of sentences used, after splitting run-on sentences into no more than three clauses. ‘Instructions’ is the mean total number of instructions written, across the group of route instructions.

Table 4.4 shows the most frequently used words per director. On one hand, this shows common words in the domain, best illustrated by the ‘All’ line giving the most frequent words across the full corpus. At least one of these most frequent overall words

³In one case the director started over due to an experimental error. In another case, the director asked to explore the environment further after entering in some instructions. In both cases, both in Experiment 1, the director gave some pairs of instructions for the same routes.

Group	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
All	you	to	turn	left	right	go
Corpus 1	to	turn	left	forward	hall	right
Corpus 2	you	go	left	right	turn	to
Corpus 3	you	is	right	left	to	position
Discrete	to	turn	you	left	go	right
Male	turn	to	you	left	right	forward
Female	you	to	go	hall	intersection	right
<i>Corpus 1</i>						
EDA	walk	turn	forward	once	left	right
KXP	go	make	area	to	left	hallway
WLH	to	move	turn	left	right	is
EMWC	hall	to	intersection	go	forward	turn
KLS	path	take	at	go	intersection	right
TJS	hall	then	down	is	go	with
<i>Corpus 2</i>						
JJL	to	you	left	continue	untill	now
JXF	go	forward	you	until	turn	wall
MXM	go	left	take	right	you	until
MJB	you	hall	go	down	take	left
PXL	move	you	floors	with	direction	position
QNL	across	panel	turn	move	right	left
BKW	walk	you	take	on	with	to
BLO	go	make	down	stops	right	stop
JNN	go	you	turn	hit	until	left
LEN	intersection	hall	is	position	it	in
MXP	forward	move	intersection	take	you	floors
TXG	turn	move	forward	then	you	right
<i>Corpus 3</i>						
JTM	forward	turn	move	stop	hallway	once
KAJ	hall	you	until	walk	down	intersection
KXK	section	tile	take	go	end	at
MHH	go	left	take	right	to	blue
RRE	hall	blue	until	position	reach	you
WAB	hallway	take	at	left	stop	intersection
ARL	you	on	in	position	is	floors
JLM	you	walk	is	on	position	at
JXL	hallway	walk	brick	is	at	intersection
LCT	floors	to	you	position	right	will
SCD	corridor	you	walk	make	until	on
SMA	is	to	road	your	has	position

Table 4.4: Most frequent words per group of instructions. ‘Word 1’ is the most frequent for that group, ‘Word 6’ is the 6th most frequent word. Lines indicate sets of groups: All, by experiment, both sets of discrete motion experiments, by gender, and by director from each of the three experiments.

Group	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6
Corpus 1	alley	hatrack	containing	octagon	alleys	rose
Corpus 2	1	panel	stops	rectangles	panels	eisle
Corpus 3	run	arrive	turtle	board	walking	couch
Discrete	segment	tiled	alley	segments	hatrack	now
Corpus 3	run	arrive	turtle	board	walking	couch
Male	alley	untill	head	panel	octagon	rectangles
Female	tiled	bare	containing	blank	passing	stops
<i>Corpus 1</i>						
EDA	walk	once	twice	times	so	back
KXP	area	make	aesal	rt	keep	going
WLH	alley	move	carpet	octagon	flooring	alleys
EMWC	segment	tiled	segments	bare	containing	passing
KLS	path	take	towards	plain	coat	cement
TJS	rose	grey	take	movement	middle	winding
<i>Corpus 2</i>						
JJL	untill	come	continue	now	sure	patterned
JXF	easel	gray	road	path	dark	paths
MXM	poster	make	posters	1st	know	between
MJB	rectangles	1	will	makes	eisle	segment
PXL	step	find	tunnel	place	stand	crossing
QNL	across	panel	panels	cross	hex	times
BKW	walk	them	scales	interrection	bricks	u
BLO	stops	make	stand	painting	three	over
JNN	available	tiling	hallway	square	hanger	closest
LEN	sitting	grey	tiled	floral	easle	dark
MXP	puke	past	twice	stone	set	degrees
TXG	re	foward	now	hallway	hanger	than
<i>Corpus 3</i>						
JTM	move	forward	once	twice	face	flooring
KAJ	start	halls	blocks	intersection	easle	block
KXK	section	tile	head	destination	tiles	arrive
MHH	till	last	hit	stand	off	bricks
RRE	area	follow	find	army	orange	pink
WAB	painters	hat	continue	intersection	stool	stand
ARL	located	corridor	floors	blank	space	standing
JLM	easel	path	pictures	granite	intersection	four
JXL	post	gray	intersection	flower	artist	hanger
LCT	floors	turtle	rock	shells	shell	locate
SCD	corridor	make	colors	onto	intersect	plain
SMA	road	exits	has	big	eiffel	honeycomb

Table 4.5: Most frequent words per group of instructions. These are the words with the highest text frequency-inverse document frequency ratio. These words occur frequently in the listed instruction group, and seldom occur in other groups. Lines indicate sets of instructions: by experiment, by motion type, by gender, and by director from each of the three experiments. Typos are by the directors.

occur in the top six words of each director.

On the other hand, the most frequent word list also shows the variation among directors. The different directors use different motion verbs (e.g. *turn* vs. *take* vs. *make*), take different strategies in terms of landmark description (e.g. *intersection*, *red*, *tile*) or purely action description (e.g. director EDA). Some directors use a variety of verbs often (e.g. *hit*, *stop*, *find*, *reach*), while other directors only use simple *turn* and *travel* verbs and *is* frequently. Even some misspellings crop up in the most frequent words (e.g. *untill*).

Table 4.5 conversely shows the most distinctive words for each group, compared to its peer groups. This is calculated by the text frequency inverse document frequency ratio (TFIDF). Here the ‘documents’ being considered are the concatenation of all instructions by a director or group of directors. Words are weighted by their frequency within the instructions from the group of directors, then normalized by the inverse ratio of how many of the peer groups the word appears in. These words are frequently used by one group, but used by few others in the peer group. This table highlights the diversity of the directors in vocabulary and concepts used, as well as creativity in spelling (There must be fifty ways to spell *easel*).

4.7 Human Followers Read, Navigate, and Evaluate

All three experiments used the same procedure for people to follow and rate route instructions. This section describes the general procedure, then the design and participants of the follower experiments.

4.7.1 Procedure

Followers are acclimated to navigation and experiment cues and then follow sets of route instructions from all directors in all environments.

Route Instruction Trial Procedure

For each route, followers were shown the instructions in a pop-up window (Figure 4.3). The navigation screen start was initially blank, with the instruction window is in front. Followers could look over the instructions for an unlimited time. Also, followers were told they they could re-examine the instructions at any time while navigating, so they did not need to memorize the instructions. When the followers were ready, they clicked the 'OK' button, which closed the text window.

The follower was placed at the starting position facing one of the four directions (randomly selected). The virtual curtain was removed and the followers could see the navigation screen. The navigation control was identical to the control for the directors (Section 4.3.1).

The followers never heard the named position announcements as the directors did. The followers navigated and recognized the destination from the text in the current instructions alone. By pressing the 'd' key, the follower could review the instructions at any time while navigating, but the follower had to close the instruction pop-up window to resume way-finding. This allowed us to measure when the followers referred to the instructions.

When the followers reached the destination or finished trying, they pressed the space bar. Each trial continued until the follower explicitly indicated completion, either when they found the destination or when they gave up. The followers were instructed that they should use common sense to follow the instructions, as they did contain some errors.

After terminating, the followers were asked the same rating questions the director used in self-rating (Figures 4.4 and 4.5): "How certain are you that you are at the destination?" "How good do you feel the directions were for getting to the destination?" Each question was answered on a Likert scale from 1 (poor instructions/ certain not there) to 6 (excellent instructions/ certain at destination).

Route Instruction Trial Sequence

After rating the instructions, the followers repeated the same procedure for the next set of instructions. The followers followed one set of route instructions for each route in all three environments.

The sequencing of the route instructions was designed to discourage the followers learning the environments or any particular director's style. The followers changed environments every other route and had no direct indication of which environment they were in.

The instruction sequences were also constrained by the directors and endpoints of the routes. No follower ever navigated exactly the same route twice. No director was repeated within any four instruction sets. No position was repeated as a start or destination within 3 trials.

After reading, following, and rating 126 routes, or when their time period has elapsed, the follower was finished participating in the experiment.

4.7.2 Experiment 1: 24 people following 6 directors' instructions

Experiment design

In Experiment 1, the followers each followed instructions from all six directors from Experiment 1 in all three environments. The sequence of instructions is detailed above.

Participants

Participants are drawn from the pool for the introductory psychology class at the University of Texas at Austin, and receive one or two hours of course credit for their participation. The participants range in age from 19 years to 29 years (mean 20.1; median 20). By design, there were equal trials by males and females.

Route instruction preparation

In Experiment 1, the instructions were exactly as typed by the director, including any typos and line breaks. The only change was anonymizing the position names, to prevent the followers from relying on the names of the positions. Only the numerical identifier was changed, replaced with a single letter, ‘X’ for destinations and ‘Y’ for starting positions. For instance, “That is Position 3” became “That is Position X” and “From Pos-2, ...” became “From Pos-Y, ...”.

4.7.3 Experiment 2: 44 people following 18 directors’ instructions

Experiment design

In Experiment 2, the followers followed instructions from the six directors from Experiment 1 and the twelve instructions from Experiment 2. The instructions were intermingled so that each follower followed and evaluated both Experiment 1 and Experiment 2 instructions. This measures if followers in Experiment 2 follow or rate the instructions differently than the followers in Experiment 1.

In Experiment 2, the followers were switched between six sets of named positions, two in each of three environments. Following a later set of instructions in the same environments, the follower may later proceed along a route to a different “Position 2.” The followers were warned that they would be switched among maps and position sets and to follow the instructions independently.

Participants

Participants are drawn from the pool for the introductory psychology class at the University of Texas at Austin, and receive one or two hours of course credit for their participation. The participants range in age from 18 years to 25 years, 7 months (mean 19.8; median 19.5). By design, there were equal numbers of males and females.

Route instruction preparation

For Experiment 2, the instructions presented to the followers were cleaned up for the parser. Run-on sentences were segmented into no more than three independent clauses per sentence and consistently punctuated with periods. Typos that split or joined words were fixed, though typos within words remained. The instructions were presented with each sentence starting on a new line. In this version, the position names were not altered, but subjects were told that the names are not consistent between trials.

4.7.4 Experiment 3: 24 people following 12 directors' instructions

Experiment design

Participants

Participants are drawn from the pool for the introductory psychology class at the University of Texas at Austin, and receive one or two hours of course credit for their participation. The participants range in age from 18 years, 1 month to 29 years, 7 months (mean 20.2; median 19.3). By design, there were equal numbers of trials by males and females.

Route instruction preparation

For Experiment 3, route instructions were prepared as for Experiment 2.

4.8 Human Task Performance Overview

Figure 4.7 shows how the mean success rate of human followers varies by the mean post-hoc human rating, over instructions from all three Experiments, by followers from all three experiments. The success rate is merely how often the follower reached the destination to which we asked the director to guide the follower. There was a very strong correlation between the mean success rate and the mean subjective rating – Spearman

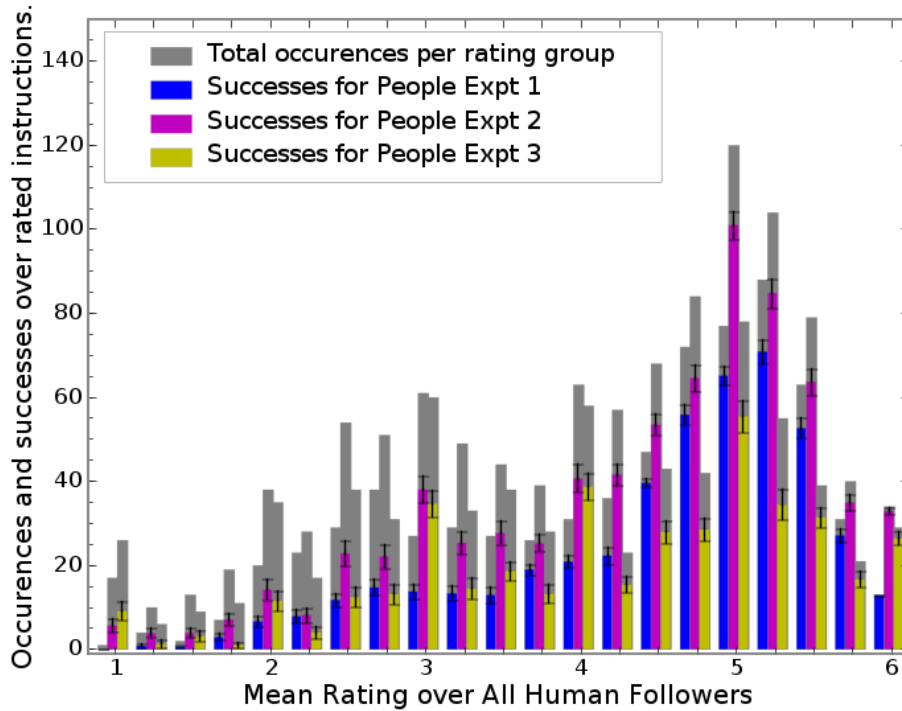


Figure 4.6: Occurrence and success rates over all human followers by mean rating. Most instructions are good to excellent, with the median rating at 4.0 and the mode at 5.0. Gray bars indicate the total number instructions with that mean rating in that corpus, colored bars the number of successes of people following those instructions.

$r(19) = 0.957, p \leq 0.001$. Previous work has also found an significant relationship between instruction quality and instruction following performance (Daniel et al., 2003; Denis et al., 1999).

For breakdown of instructions by linguistic constructs used and spatial reasoning needed, see Chapter 6 and Appendix B.

Figure 4.6 shows the distribution over post-hoc human rating, and success rates for followers from Experiments 1, 2, and 3. The followers from Experiment 2 have a very slight performance increase over the followers from Experiment 1 on the instructions from Experiment 1, ($M = 70\%$) vs. ($M = 69\%$), $t(638) = 2.46, p \leq 0.014$. The followers

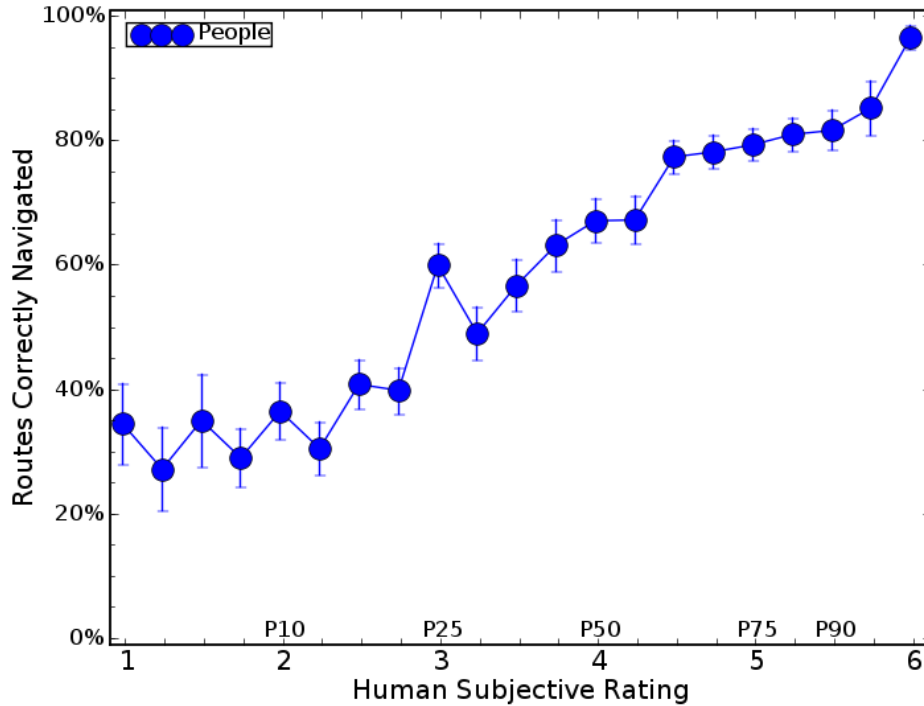


Figure 4.7: Mean performance over all human followers. The cumulative success rate (Y-axis) over all instructions from all three experiments by mean post-hoc instruction rating (X-axis). Success rate is how often followers finished at the intended destination for instructions with mean rating of $r \pm$. Data as of May 31, 2007. The annotations P10... P90 indicate the 10th ... 90th percentiles of ratings.

from Experiment 3 only followed the instructions from Experiment 3, as both groups used the continuous motion controlled by the joystick, whereas all people in Experiments 1 and 2 used discrete motion by the keyboard. The subjective ratings were not evenly distributed across instructions. Most instructions in this corpus were rated highly; the mode rating is 5.0. In the middle rated instructions – rated between 2.0 and 4.5 inclusive – the distribution was fairly even.

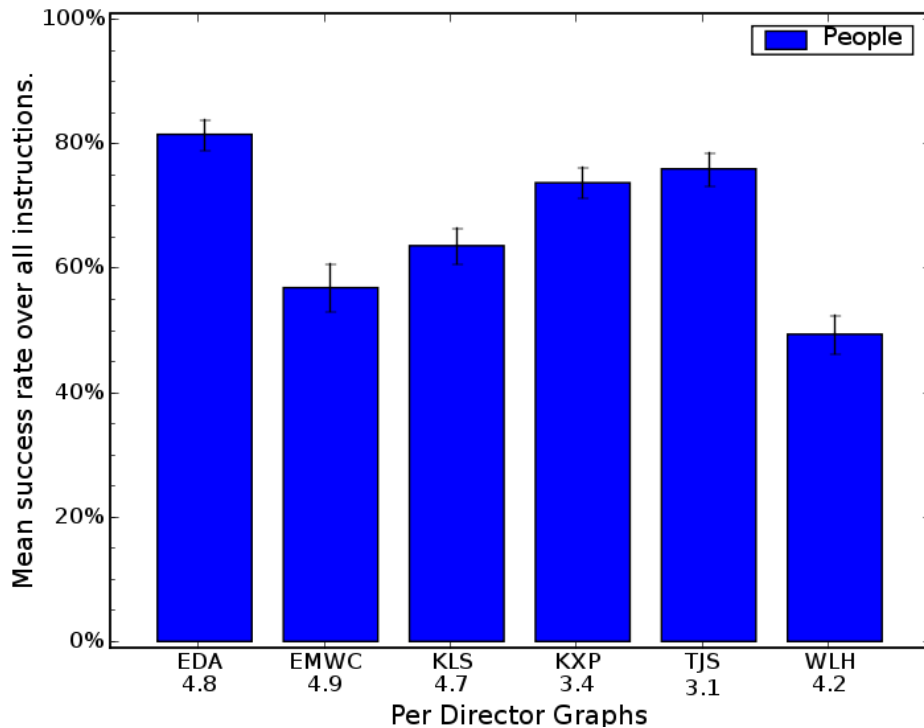


Figure 4.8: Success rates over all human followers per Corpus 1 director. The vertical bars show the success rate per director by followers from Experiment 1. Under the bars are the directors’ initials and the mean subjective rating (1-6 Likert scale) over all the directors’ instructions by all followers from all three experiments.

4.8.1 Differences in Directors

There were strong individual differences among the directors. On human follower performance, directors ranged from 83% successful instructions to 31% successful, even after the blank instructions were filtered out. The mean of the 30 director’s mean performances was ($M = 60.8\%$, $SEM = 2.6\%$), while the mean success rate over all 1517 followed instructions was ($M = 64.5\%$, $SEM = 1.0\%$). The directors who gave better instructions also tended to give more non-blank instructions, skewing the overall mean up from the mean of directors’ means.

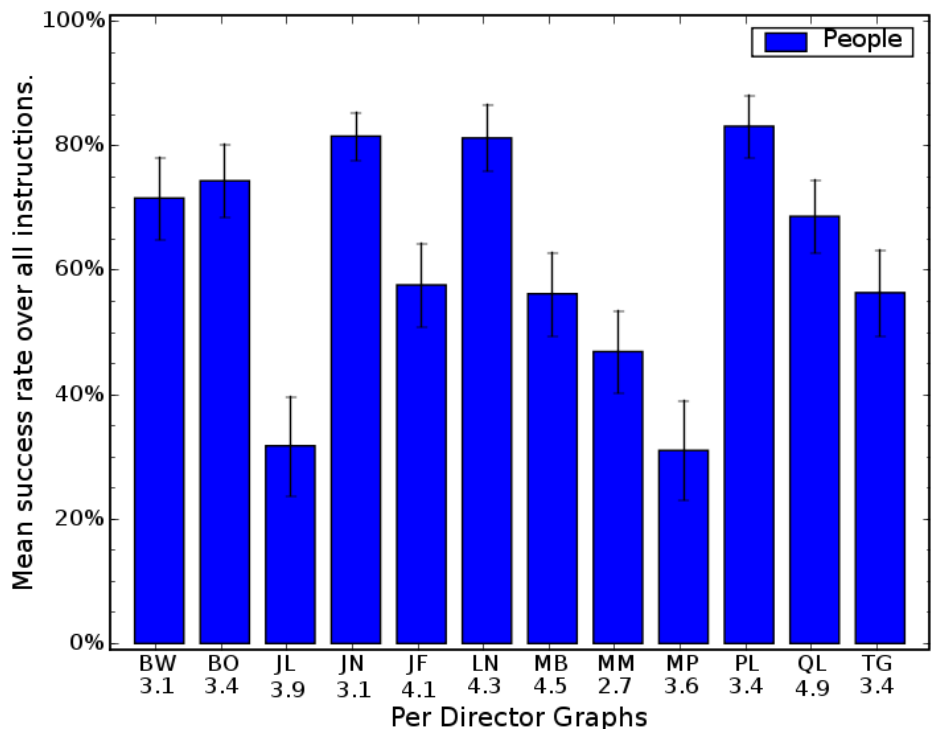


Figure 4.9: Success rates over all human followers per Corpus 2 director. The vertical bars show the success rate per director by followers from Experiment 2.

Are the differences between the directors' success rates significantly different? To answer this question accurately, we need to control for the different sets of start and target position pairs across directors. In Experiment 1, the directors were asked to describe the same 126 routes in all three environments, but some did not write instructions for each route. In Experiment 2 and 3 the directors each were asked for instructions describing 42 routes in one environment. Moreover, half the later directors described routes with different starting and ending locations than the Experiment 1 directors. Since the routes vary in complexity and difficulty, a fair comparison controls for the routes described.

Assume a director d , wrote instructions RI for routes R through one or more environment. We can collect the instructions RI' from all other directors d' describing these

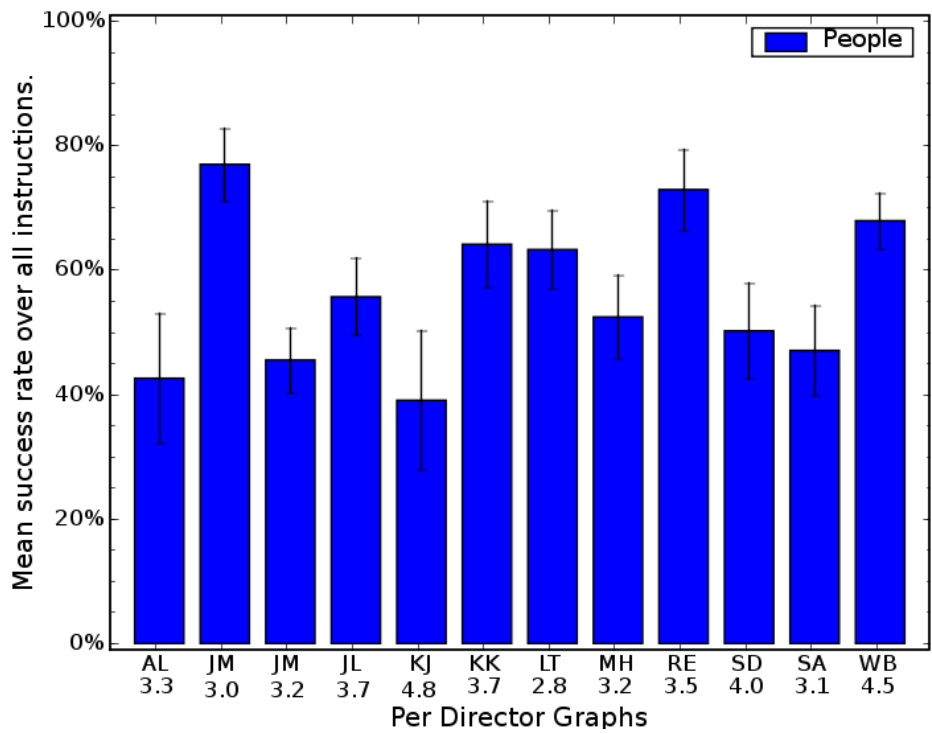


Figure 4.10: Success rates over all human followers per Corpus 3 director. The vertical bars show the success rate per director by followers from Experiment 3.

same routes R . We compare the mean performance of the human followers on the two sets of instructions (RI and RI') describing the same routes R . This controls for the differences in complexity of the routes. We performed a two-tailed paired t-test on the mean success rate of all human followers to determine if the instructions RI from a director d differed in success rate from the instructions RI' for the same routes R from all other directors d' .

24 of the 30 directors had a statistically significant difference in the success of the human followers on their instructions, compared to the instructions for the same routes from other directors. Figures 4.8, 4.9, and 4.10 show the success rates for followers from all three experiments on instructions from directors from all experiments respectively. Note that followers from the first experiment did not follow instructions from Experiment 2,

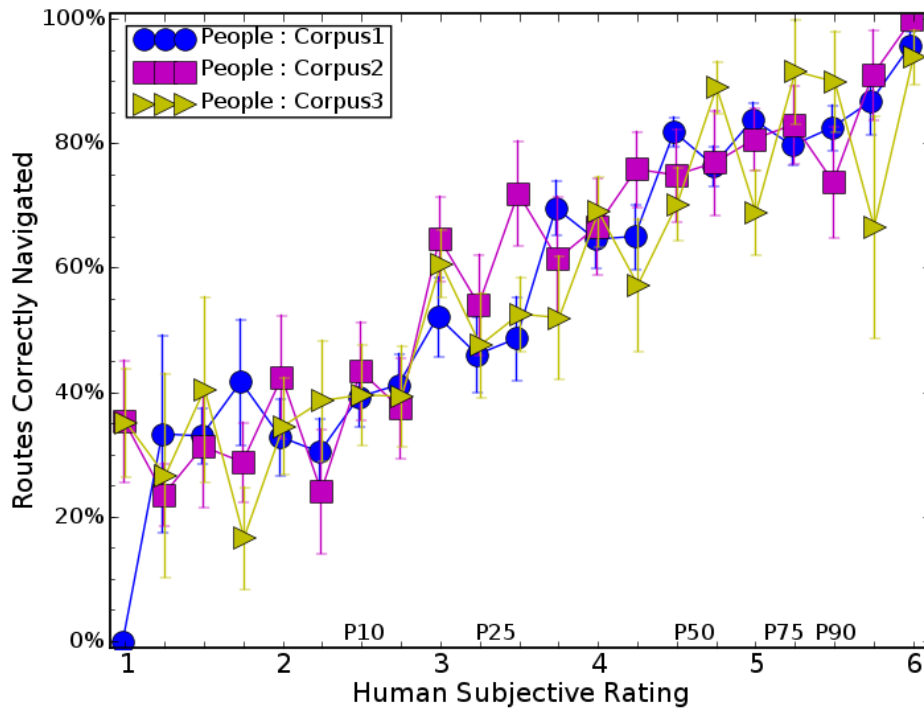


Figure 4.11: Success rates for all human followers from each experiment by instructions rating. There were no significant differences in how people followed the instructions from Experiment 1, 2, and 3 across the quality spectrum.

but followers from Experiment 2 did follow instructions from Experiment 1. There are significant differences in the success rates of different director's instructions, which are reproducible given different people following the instructions.

4.8.2 Differences in Human Followers between Corpus 1 and Corpus 2

Figure 4.11 shows the performance of human followers from the two experiment on instructions from each experiment. In Experiment 1, the followers reached the intended destination on with a mean of 68.5% (691 instructions, $SEM = 1.3\%$). Followers in Experiment 2 following instructions from Experiment 1 succeeded on a mean of 69.7%

(639 instructions, $SEM = 1.9$). This difference is small and barely significant, $t(638) = 2.46, p \leq 0.014$.

We controlled for route complexity by comparing the success rates on all route instructions describing routes with the same starting and ending positions. When the routes followed are controlled for, the difference disappears entirely, each at 68%. This is the mean success rate of all the followers for all the instructions describing all routes with the same start and destination. Comparing the mean success rate by route controls for the complexity of the route across directors and followers.

The 432 instructions from Experiment 2 were slightly worse, with the followers from Experiment 2 succeeding only with mean 64.0% overall ($SEM = 1.9\%$), and 66% on the routes shared with the corpus from Experiment 1 (187 instructions, $SEM = 0.6\%$, $t(834) = 5.60, p \leq 0.001$). This difference is significant from the success rate of Experiment 2 followers on Experiment 2 instructions, $t(194) = 5.62, p \leq 0.001$, controlling for route. The best directors is Experiment 2 performed at the level of the best in Experiment 1, as can be seen in Figures 4.8, 4.9, and 4.10.

4.8.3 Gender-linked performance differences

For directors, there is a strong effect of gender (Figure 4.12). For the 720 instructions from male directors, the mean success rate for all human followers was 66.5% ($SEM = 1.4\%$), but for the 797 instructions from female directors, the mean success rate for all human followers was 62.6% ($SEM = 1.3$), significant with independent $t(1518) = 2.02, p \leq 0.022$. On the common routes, the mean performances of male directors ($M = 67.4\%$, $SEM = 0.8\%$), female ($M = 62.5\%$, $SEM = 0.8\%$), significant at $t(794) = 4.79, p \leq 0.001$. See the prior section for a discussion of the mean success rate per route statistic. For followers, there was a small, but significant effect of gender: for all 1363 instructions followed by both genders, the mean success rate per route was 65.5% for male followers ($SEM = 0.5\%$) and 64.5% for female followers ($SEM = 0.5\%$), different

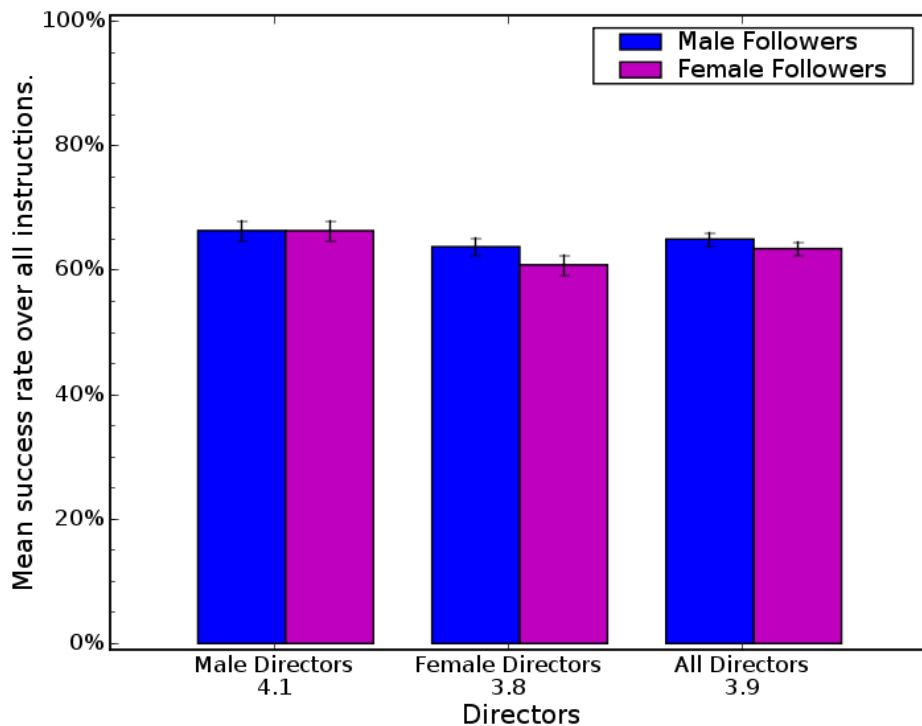


Figure 4.12: Success rates for human followers and directors by gender. Under the bars are the mean subjective rating (1-6 Likert scale) over all the directors' instructions by all followers from all three experiments.

at $t(1362) = 1.99, p \leq 0.024$.

There are not strong interaction effects of the genders of the directors and followers. On the 637 instructions from male directors, male followers have a mean success rate per route of 66.5% ($SEM = 0.9$); female followers 67.2% ($SEM = 1.0$), not a significant difference $t(636) = 0.70, p \leq 0.485$. Likewise, for 712 instructions from female directors controlling for route, male followers succeeded 63.9% ($SEM = 0.9\%$) and female followers succeeded 61.5% ($SEM = 0.9$), significant at $t(712) = 2.71, p \leq 0.007$.

Regardless of the gender of the follower, people following instructions from female directors reach the intended destination 6-10% less often, across the three corpora. Male

followers of men’s instructions ($M = 68.2\%$, $SEM = 0.9\%$) vs women’s instructions ($M = 63.3\%$, $SEM = 0.9\%$), $t(719) = 3.51, p \leq 0.001$. Female followers also succeeded significantly more often on men’s instructions than other women’s instructions on the same routes: ($M = 68.0\%$, $SEM = 1.0\%$) vs. ($M = 61.5\%$, $SEM = 0.9\%$), $t(708) = 5.16, p \leq 0.001$.

Previous studies have also found that women are less accurate in giving instructions, particularly when the director is describing the route from memory (“map-not-present condition”) (Brown et al., 1998; Ward et al., 1986). Both of these studies, accuracy is measured by an expert rating whether the director made a crucial error of omission (implicit action) or commission (explicit mistake). Some other researchers have also found women have lower success rates than men in following route instructions (Allen, 2000), while others did not find this effect (Schmitz, 1999).

4.9 Discussion

This chapter presents a series of experiments gathering a large language and task corpus of spatial route instructions. These experiments were designed to gather instructions for complex spatial tasks using a methodology that will elicit natural variation in the instructions. The instructions vary in vocabulary, grammar, style, spatial actions and landmarks, and explicit errors and implicit omissions. Because multiple directors describe each route and multiple people follow each set of instructions, the corpus measures human variability in generating and understanding spatial procedural instructions. This corpus is unique in tying together many participant-written instructions to multiple independent human follower action traces and subjective ratings for each route instruction text.

For the directors, the methodology ensures no *a priori* knowledge of the environments, measures how quickly and how well the directors learn to navigate an environment from a first-person perspective, and gathers many route instruction texts for routes throughout the environment. Directors plan and describe the routes from

memory, a more cognitively challenging task than planning while looking at a map or other environmental model. The directors describe the routes in a paragraph of text that can be independently followed by multiple human or software followers.

The routes described in these experiments require complex procedures to navigate through large-scale spaces – buildings where the destination is not visible from the starting location or reachable with one travel action. The task is similar to that faced by patients in large hospital complexes, by new students on in large buildings or campuses, and, outdoors, by people navigating through homogeneous housing sub-divisions, where all the houses look the same and the hard to read street signs mark similar names.

For the human followers, the experiment collected independent action traces and subjective evaluations from several people following each set of instructions. This measures the behavior of different people in response to the same route instructions. Followers did not experience the environments outside of following instructions and they were shuffled between environments to discourage learning the maps. Each follower of a particular set of route instructions was presented the same route instruction text in the same manner in the same environment. Multiple human followers allow exploration of individual differences in spatial and language ability, strategy used, and even luck when presented truly ambiguous instructions. Thus the experiment measures how well any set of instructions works for guiding arbitrary, unknown followers through the large-scale space.

We measured large and statistically significant differences in both directors' route instruction verbal styles and the success rates of their instructions. This experiment measures a high correlation between subjective ratings of route instructions and the route instructions' success rates. The tasks are challenging; the mean success rate across all followers is 64.3% (1522 instructions, $SEM = 0.9\%$).

We measure the statistical distribution of route instruction quality across directors, environments, and route complexity. The median instruction is ranked 4, with 6 being excellent, but there is a long and fat tail of poorly rated instructions, with 25% of the

instructions rated between 3 and 4, and the final 25% rated between 1 (the worst rating) and 3. Put another way, for these environments, with directors trained until they could navigate efficiently, one quarter of the instructions are excellent, one quarter good to very good, one quarter medium to good, and one quarter poorly rated, with objective success rates matching the subjective ratings. Across the different experiments, the same patterns occur in the data in the distribution of instruction quality and the strong correlation between quality and success rate.

This route instruction language and task corpus gives insight into open questions in the literature. For instance, by splitting directors and followers by gender, we find both male and female followers succeed about 10% more often given instructions from male directors. There is a smaller, but still significant, gender-linked difference in the follower's success rates: female followers succeeded about at about the same rate as male followers on instructions from male directors, but about 3% less often than men on instructions from female directors. However, with the large variation in success rates, some of the women directors had very high success rates, and some of the male directors very poor success rates. Moreover, we have not yet done an analysis controlling for experience playing first-person video games, which may partially explain these differences.

This chapter describes a human cognitive psychology experiment into how people learn, reason, and describe complex routes through large-scale spaces. We collected a large language and task corpus of directors learning the environments and giving instructions. We also collected multiple followers applying each instruction text to perform the task and evaluating the instructions. From these experiments, we gain a better understanding of the variability in instruction-giving and instruction-following. Finally, the corpus shows us the distribution of route instruction quality and styles that an instruction follower must handle to robustly follow natural instructions.

Chapter 5

Spatial Route Instructions in the MARCO Architecture

5.1 Understanding and Following Route Instructions in Context

MARCO is an architecture for understanding and following natural language route instructions (MacMahon et al., 2006). MARCO is composed of six modules: three modules interpret the route instruction text linguistically; three modules interpret the instructions spatially in the context of the task and environment.

MARCO's linguistic modules parse raw text and produce an imperative procedural model – a skeletal plan. The *syntax parser* models the surface syntactic structure of an utterance. The *content framer* abstracts away from arbitrary word order and formation, to model the surface meaning of the utterance. The *instruction modeler* applies spatial and linguistic knowledge to combine information across phrases and sentences. Figure 5.1 shows the representations MARCO uses to model route instructions.

MARCO's executive modules apply the instructions to navigate through the world. The *executor* reactively interleaves action and perception, acting to gain knowledge of the environment and execute the instructions in the context of this spatial model. The *robot*

controller is an abstraction layer for particular robots' motor and sensory capabilities. The *view description matcher* checks symbolic view descriptions against sensory observations and world models – checking the expected model against the observed model. Figure ?? shows the relation of the natural language understanding and robot control parts of the architecture.

The language understanding part of the architecture builds on ideas from the Nautilus natural language understanding system (Perzanowski et al., 2001; Simmons et al., 2003; Wauchope et al., 1997), from Reiter and Dale's natural language generation architecture (1997), as well as frame-based systems (Bindiganavale et al., 2000; Chang et al., 2002). The instruction execution engine follows in the tradition of reactive execution or sequencing code in the middle tier of three-tiered intelligent architectures (Bonnasso et al., 1997; Firby, 1989; Verma et al., 2005), with a prototype implementation programmed in TDL (Simmons and Apfelbaum, 1998) (See Section 2.4.3).

5.2 Syntax Parser

The *syntax parser* parses the raw route instruction text. Our implementation uses a probabilistic context-free grammar built with the Python Natural Language Toolkit (Bird and Loper, 2004). Instead of modeling part-of-speech syntax, our grammar directly models verb-argument structure, similarly to (Baker et al., 1998; Bindiganavale et al., 2000; Chang et al., 2002; Palmer et al., 2005). The top of Figure 5.1 provides an example of a parse tree.

The verb-argument grammar is detailed in Appendix C.2. We aim for MARCO to understand the language of route instructions well enough to navigate. MARCO is not assisted by having a correct part-of-speech parse tree of a sentence without the semantics of the words and phrases. Since MARCO needs a strong model of the verbs, adjuncts, and referring phrases in the domain to execute the instructions, it does not hurt to restrict the grammar to the domain. Moreover, this approach can model which arguments are optional and which required for different verbs. Finally, MARCO attempts to execute instructions

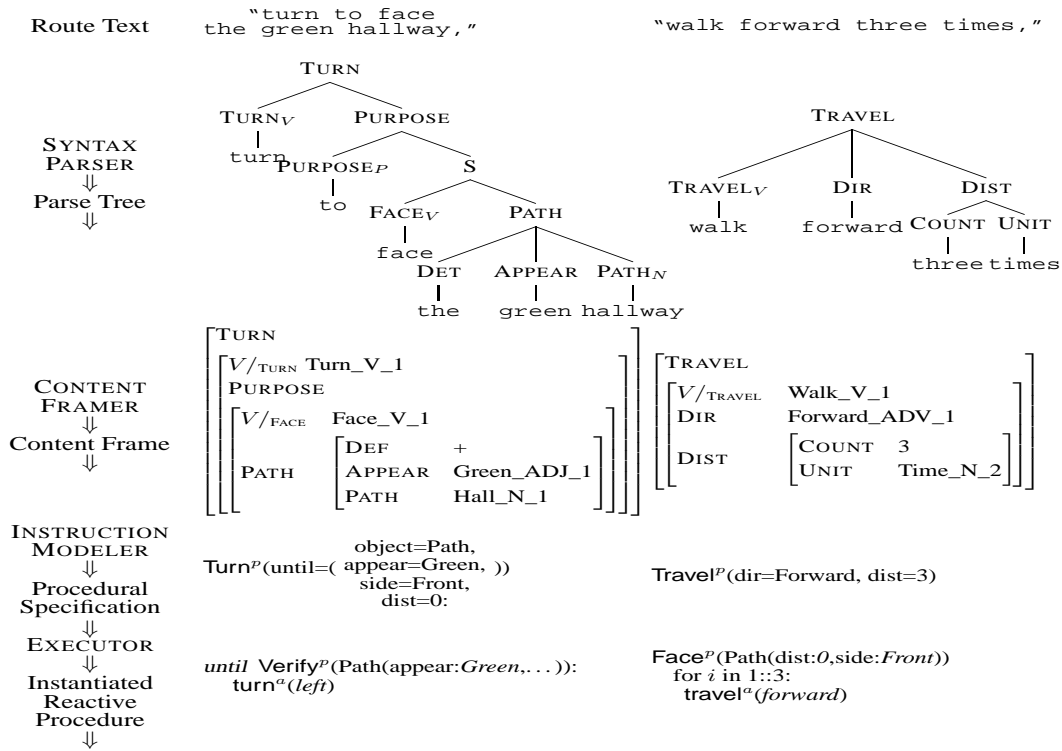


Figure 5.1: MARCO linguistic modules modeling a route instruction text (Top) through the syntactic verb argument and phrase structure (Mid-Top), the surface semantics frame (Mid-Bottom), and the imperative semantics of which procedural to take, given the context (Bottom).

even when it does not understand some sentences or words. Since much of the complexity of instructions can be in elaboration sentences describing extra details, this is a viable strategy.

The parser in the implementation is sufficient for to investigate the issues of this thesis, although future work should look at integration with best-of-breed parsers. A more traditional part-of-speech grammar and parser would require a more complex surface semantics interpretation module. One interesting approach would combine a re-ranking probabilistic parser (Charniak and Johnson, 2005) with statistical semantic role labeling (Gildea and Hockenmaier, 2003; Gildea and Palmer, 2002). The re-ranking could include semantic well-formedness and task and environment fit measures; Purver et al. (2006)

discuss a similar approach in interpreting single commands in a restaurant recommendation system domain. Another recent approach directly learns to translate text to a semantic representation (Ge and Mooney, 2005; Kate et al., 2005; Wong and Mooney, 2006).

5.2.1 Training the Probabilistic Context-Free Grammar

A *referring phrase* is a noun phrase that refers to some entity or attribute being described, analyzed on its semantic content instead of its syntactic makeup (Kuipers and Kassirer, 1987; Reiter and Dale, 1997). By tagging the referring phrases and verbs in a set of route instructions with semantic annotations to the phrase and argument types, this analysis characterized the surface meaning of route instruction utterances.

From the hand-labeled text, a Probabilistic Context-Free Grammar (PCFG) was trained to parse and semantically tag route instruction texts. This parser labeled additional route instructions. The automatically tagged route instructions are then hand-corrected. The tagged trees help bootstrap the process of learning the full language model. The architecture can either parse raw input text using the trained parser or can load hand-corrected gold-standard parse trees.

5.2.2 Robustness for the Syntax Parser

Several techniques are used to provide the syntax parser with robustness to out-of-vocabulary words and out-of-grammar utterances. The parser uses a maximum likelihood function to label novel words or familiar words used in a different linguistic context than previously seen. When an unknown word is found, the grammar matches the rest of the utterance, with a place holder for the unknown word. The likelihood of a non-terminal production yielding UNKNOWN is weighted by the variance of the non-terminal.

Open set words, such as nouns and adjectives, tend to have a large number of different tokens, while closed set words such as prepositions have very few (even counting typos separately). Thus, the grammar can guess the role of an unlearned word based on

the utterance context and the variance of the different non-terminal tokens. For instance, given “Turn onto the grue hall,” the grammar will correctly guess that `grue` is an appearance term modifying the `PATH` phrase.

When the grammar cannot find a parse for an utterance, the syntax parser will skip it. These measures prevent the processing of a paragraph of instructions from being derailed by one unknown word or grammatical construction. Often in route instructions, the more complex constructions are elaborations not strictly necessary to follow the instructions, especially if the follower is robust to gaps in the instructions. Of course, in some cases, this will leave MARCO unable to interpret the instruction correctly, but then people are not always able to interpret an instruction correctly either. Better to push forward with the well-understood part of the instructions than to attempt to integrate a poorly modeled utterance.

5.3 Content Framer

The *Content Framer* translates each utterance’s surface structure to a model of the surface meaning: a recursive attribute-value matrix that makes content readily accessible. The resulting *content frame* (see middle of Figure 5.1) models the nested structure and sense of an utterance by dropping punctuation, arbitrary text ordering, inflectional suffixes, and spelling variations.

The Content Framer looks up nouns, verbs, adjectives and adverbs in WordNet (Fellbaum, 1998; WordNet, 2005), an ontology for English. This process abstracts over surface differences in morphology, spelling, and synonym usage. The WordNet ontology is used to look up the nearest known synonym or more abstract hypernym to each mentioned unknown word. MARCO can substitute a general concept, such as “couch”, when it does know a sub-type, such as “futon.” This matching is currently a simple lookup for the nearest WordNet synonym set in a small domain dictionary, based on the lexical item and semantic role tagging in the parse tree.

From a semantic parse, as implemented in MARCO, the content frame is a small

step. Having the two processes separate eases any later integration of traditional part-of-speech parsers. The content frame provides an information interface to the rest of MARCO, so changing the parser (e.g. to a statistical re-ranking parser or another semantic parser, as noted above) only involves altering the content framer. The final reason the content framer is separate is that it is simpler to correct the semantic parse trees and infer the word sense information than to hand-correct the word sense information directly.

5.3.1 Robustness to unknown words and sentence structures

When MARCO comes across a word that it does not have in its concept base, it searches for the nearest known synonym or more abstract hypernym using the WordNet ontology. For instance, when instructed to “face the futon,” MARCO will discover *futon* is not in its concept base, look it up in WordNet, find the broader concept of *couch* in its concept base, and stop turning when the view description matcher observes a *couch*.

MARCO is also robust to unexpected input. If the content framer encounters a constituent that it cannot model, it will ignore it while modeling the remainder of the clause. Likewise, if the parser cannot parse one sentence from a set of route instructions, it will parse the others. These techniques work well for two reasons. First, route instructions often contain a lot of redundant information, so neglecting to understand a phrase in one sentence is often not critical. Second, the essential information in route instructions is usually stated using a relatively small variety of verb frames for directing movements. Most of the novel sentence frames occur in the declarative descriptions between movement commands, functioning as elaborations. These are often not necessary if the imperative sentences are properly understood and applied.

5.4 Instruction Modeler

The *instruction modeler* translates the *content frame*'s representation of the surface meaning of an instruction element to an imperative model of what to do under which conditions –

the *procedural specification*. The instruction modeler infers the imperative model from the instructions by applying linguistic knowledge of the verbs and prepositions of the route instructions and spatial knowledge of how perception and action depend on the local spatial configuration in similar environments. Each verb frame is associated with a hand-coded procedure to build a procedural specification from the content frame, based on recognizing frame arguments and idioms.

The instruction modeler integrates content frames into a model of the entire instruction set. The *imperative* information in the instructions – what to do – modeled as *procedural specifications*. The *declarative* information in the instructions – what to expect – is modeled as *view descriptions*. The procedural specifications use view descriptions to model the constraints on when to execute causal actions.

The *instruction modeler* builds a skeletal plan from the instructions by applying two kinds of knowledge. The follower needs both general knowledge about language – especially common imperative constructions – and domain-specific knowledge of the verbs, prepositions, and referring phrases of spatial route instructions. Prepositions and verbs of motion in route instructions are relatively independent of where instructions are followed (the environment) and who follows them (the agent). On the other hand, perceiving and acting on objects and object attributes are dependent on the agent and environment.

The instruction modeler interprets referring phrases, declarative statements, and implied landmarks (such as a path for a travel verb) as view descriptions to be matched while executing the task against direct observations of views and synthesized models of scenes and the state of the instructed task. The instruction modeler interprets both the imperative commands and the declarative descriptions in the instructions as procedural specifications. A procedural specification instantiates a general reactive procedure with the verb arguments and adjuncts that are contingently achieved by other procedures. The stated and implied information about the world is modeled as view descriptions, which capture the entities mentioned and the structure of the relationships between them.

The instruction modeler also decomposes high-level commands into lower-level procedures. The concise “Take the third right to the end of the hall,” is modeled and executed very similarly as if the director had explicitly commanded, “Go down to the third place with a path to the right. Turn right there. Go down to the end of that hall.” This simplifies the execution code, by separating it from the surface form of the instructions.

Even relatively an explicit and simple command is interpreted as a reactive procedure of simpler procedures. For instance, “Take the blue hall to the chair,” may require a *Travel^p* procedure to move to the blue hall, a *Turn^p* procedure to face along the blue hall towards the chair, and the explicit *Travel^p* procedure along the blue hall until the chair is reached. The first *Travel^p* to the blue hall may, in turn, require a *Turn^p* procedure to face the blue hall and possibly a *Find^p* procedure to locate the blue hall. Each of these procedures controls the sequencing of simple causal actions – e.g. *Face^p* repeatedly executing the *turn^a* action to change orientation within a place, until the faced object is to the front.

While currently implemented by sequencing discrete, causal actions, the representation only models the conditions and the actions. Another executor could be implemented as directly calling continuously operating control laws and execution monitors for termination, as done in other related work (Lauria et al., 2002a; Simmons et al., 2003; Tellex and Roy, 2006).

The instruction modeler reasons about the semantics (meaning), anaphora (co-reference resolution), and discourse pragmatics (inferring the conversational intent of an utterance) of route instruction texts. The module encapsulates the functionality of building compositional models of utterances, combining information within and between utterances.

5.4.1 Representing Referring Phrases as View Descriptions

A *view description* represents what the follower expects at a pose in the environment, given the implicit and explicit descriptions in the discourse. The view description is a structure modeling the relations that an expected object or structural entity has with the follower or other landmarks, given the instruction text and the follower’s spatial knowledge. For each expected landmark, the view description models the object’s type, the object’s location within the view relative to the observer (angle and distance). Additionally, the view description may model any mentioned constraints on the *attributes* of an entity and *relationships* between that entity and others. For each pose, the view description may be a list of several entities without any explicit relationships among them.

The view description is a minimal model of what the follower expects: it neither over-commits to unspecified details nor enumerates all the possible worlds that would match the description. Instead, the view description hews close to what was said. For example, the *until* condition at bottom of Figure 5.1 models the post-condition of the Turn^p: the follower expects a *Path* with a *Green* appearance in front of it, but the path may be immediate in the view or off in the distance.

The instruction modeler reduces a wide variety of surface forms to functionally equivalent relational models. The module handles building view descriptions that combine noun phrases with pre-position (e.g. “the green hall”), post-position (“the hallway with the grassy floor”), predicate adjectives (“The path will have grass on the ground.”), and dependent clauses (“the hallway that has green on the floor.”). The view descriptions from these phrases will encode similar relationships about the grassy corridor being described, with slight differences in detail depending on the surface form of the phrasing. The view description matcher will not need any linguistic knowledge to match the relational models to the perception of the hallways.

Position is encoded by two attributes: *side* and *distance*. These may be relative to an intersection, the follower, or another object, from the perspective of the follower.

The default point of view is the follower for the domain of route instructions. *side* may be *Left*, *Right*, *Sides*, *Front*, *At*, *Left_Front*, *Right_Front*, or unspecified. This should later be expanded by leveraging work modeling spatial prepositions and landmark positions, especially in small-scale space, such as (Blisard et al., 2006; Herskovits, 1985; Klippel and Winter, 2005; Regier and Carlson, 2001; Skubic et al., 2004b; Talmy, 2000).

The attribute *distance* encodes the linear position front-to-back within the view. Is the object of interest immediately close, a few intersections away, or off in the distance? (Note that encoding distance in the view is a different concept than distance along a path.) *distance* values are encoded as a distance range. For instance, an immediate object is ‘0’, an object at or past the next place is ‘1 :’, an object a few intersections away might be ‘2 : 3’ and unspecified distance can be represented as ‘0 :’, but is usually not included in the view description, since it is not a constraint on the view.

This representation of view distance is implementation specific, but any route follower needs a similar representation of distance. At the least, the representation of view distance must be able to distinguish between immediate (here ‘0’) and distant (‘1 :’). These distinguish, for instance, between facing towards an object and being at the object – the pre- and post-conditions of an `until` phrase. These three view distances – immediate, distant, and unspecified – account for all but 7 of the 7,015 view distances in the procedural specifications derived from our corpus. The others come from phrases such as “one space before the chair.”

Besides positional information, arbitrary other information may be encoded as an *attribute* of a view description. Attributes model a constraint on one aspect of the entity. The current implementation models the attributes of *appearance*, *length*, *subtype*, and *count*. Attributes provide a constraint along one dimension, which may be precise (e.g. `brown wooden`) or broad (e.g. `dark`).

In addition, each entity in the view description may also model the *relationships* to other entities. Similarly to Herskovits (1985), we will call the focal entity the *subject* and

the other entities mentioned the *reference* objects. For spatial prepositions, Talmy (2000) called these the Figure and Ground objects.

In the current implementation, the modeled relationships are *Between*, *Detail* *Loc*[ated], *On*, and *Part*[-of]. *Between* is a ternary relation that the subject is located between two other reference objects, i.e. if the follower is the subject, the reference objects will be on opposite sides of her. *Detail* just asserts an unspecified connection between two objects, which may be co-location (e.g. the end with the easel) or part (e.g. the intersection with the brown hallway). *Loc* represents that the subject is co-located with (or at or in) the reference object. *On* represents that the subject is topologically *on* a reference entity, a path.

Part represents that the entity is part of the other entity, though which is the part may not be apparent from the surface form. For instance, in the end of the hall, the subject the end is part of the reference entity the hall. However, in the intersection of the red and blue halls, the subject the intersection is composed of the path fragments referred to by the red and blue halls. Thus, *Part* only models that part-hood relationship is present, leaving the view description matcher to interpret which entity is part of the other(s). The view description matching code must be domain-specific, so will have better models of the domain entities than the more general instruction modeler.

Note that some of the attributes and relationships in the view description will be from explicit mentions in the instruction text, but others will be inferred from knowledge of the task. For instance, consider the command “go down the pink-flowered hall until you reach the red-brick hall.” Here, the type and appearance of the mentioned hallways are modeled from the text: *Path*(appear:[*Rose*]) and *Path*(appear:[*Brick*]).¹

Knowledge of the *Travel*^p command is needed to fill in the pre- and post-condition

¹[*Rose*] and [*Brick*] are domain-specific symbols that the view description matcher can match in the percept stream.

distances and positions. The path traveled along will be immediate to the follower and in front at the beginning of the procedure – *Path*(appear:[Rose], dist:'0', Side:[Front]). At the beginning of the *Travel^p*, the destination path will be distant and in front of the follower (*Path*(appear:[Brick], dist:'1:', Side:[Front]), while at the end, it will be local in an unspecified position (*Path*(appear:[Brick], dist:'0')).

5.4.2 Representing Conditional Actions as Procedural Specifications

Route instructions require at least four low-level *causal actions* (Kuipers, 2000). A *turn^a* changes the agent's orientation (pose) while remaining in the same location. A *travel^a* changes the agent's location without changing orientation along a path. A *verify^a* checks an observation against a description of an expected view. A *declare-goal^a* action terminates the instruction following process by declaring the agent is at the destination. Route instructions may contain other action types, such as “open the door” or “take the elevator to the 2nd floor.” These four causal actions are both necessary to follow almost all route instructions and sufficient for many route instructions.

The *procedural specification* captures the commands in route instructions by modeling which actions to take under which external (e.g. seeing a view) or internal conditions (e.g. estimating the distance traveled). Resolving some ambiguities is deferred until the follower observes the environmental context as it proceeds along the route. These procedural specifications are similar to the Bindiganavale et al.'s Parameterized Action Representations (2000), Denis et al.'s “minimal units of information” (Daniel et al., 2003; Denis et al., 1999), Higher-Order Route Instruction Elements (Klippel et al., 2005), Spatial Routines in small-scale space (Tellex and Roy, 2006), or Navigational Information Units (Levit and Roy, 2007). Figure 5.1 shows the transformation from text to the imperative instruction model.

Each clause is interpreted as a procedural specification depending on the verb or a heuristic match based on the other constituents. Adverbs, verb objects, and prepositional

phrases translate to pre-conditions, while-conditions, and post-conditions in procedural specifications. For instance, constituents may describe which path to take, how far to travel, or the views that will be seen during the procedure. This is similar in intent to work on combining the lexical semantics resource FrameNet with action schemas, allowing inference (Chang et al., 2002).

The modeler recognizes termination conditions stated as purpose clauses (Di Eugenio, 1992), like “Turn so that you see a chair in front.” Other action verbs have some arguments implicit, such as “face” implies turn until the description is matched. Conditional procedures are modeled by embedding procedural specifications. Note the implicit travel procedure in “At the corner, turn left,” modeled as $\text{Turn}^p(\text{direction:Left, precondition :Travel}^p(\text{until:Corner}(\text{dist:'0'})))$.

5.5 Executor: Interleaving Action and Perception

The *executor* sequences causal actions given the environmental context and the state of following the route instructions. The executor interprets each procedure to execute causal actions, including verifying view descriptions to check the state of the world. For instance, given a Face^p procedural specification, the executor continues to turn^a until a verify^a returns the view description has matched.

The executor is equivalent to, and can be implemented by, the reactive task sequencing tier of the standard three-tiered intelligent architecture (Bonnasso et al., 1997). A predecessor implementation of the MARCO executor (MacMahon et al., 2004; Simmons et al., 2003) is written in TDL, Task Description Language (Simmons and Apfelbaum, 1998).

The executor algorithm need not know anything about how the follower moves through the environment or how the view descriptions are verified. Those actions and observations may be opaque to the executor stage, in an ontology it can pass through but not comprehend. Thus, an executor module may be run with differently implemented robot

controllers, controlling different hardware or software robots, with differing perceptual abilities.

The simplest route instruction executor is the *naïve instruction queue executor*. It steps through a list of instructions, attempting to execute each fully, without considering the instruction context, before moving to the next. Executing each instruction may consist of ensuring various preconditions, distance estimates, and postconditions are met. Each condition may entail moving, then verifying the resulting view matches a view description.

The MARCO implementation uses a *pragmatic instruction queue executor*, with procedures that react to both the linguistic and the spatial context. Spatially, the procedures act to achieve preconditions, for instance, a *Travel^p* procedure facing a path before moving forward. Both preconditions inherent in the procedure and specifically mentioned by the director are treated in this way. Linguistically, the procedures execute differently based on their context in the instructions. For instance, a *Travel^p* procedure is inserted in between two consecutive turns, and unterminated *Travel^p* procedures will look ahead for a destination in the up-coming instruction utterances. See Section 6.4 for an evaluation of the performance impact of considering the instruction context, as well as the spatial context, in following route instructions in our corpus.

This instruction following algorithm may be replaced with more sophisticated algorithms that leverage previous knowledge of the environment map, or knowledge of individual director's style, vocabulary, and common mistakes. The executor could also maintain more state about the route as traveled, building up a map through topological simultaneous localization and mapping (Kuipers et al., 2004). With a local map and hypothesis tracking, the executor could implement back-tracking. This would handle some mistakes that the current MARCO cannot, both the follower's errors (e.g. incorrect reference resolution) and the director's (e.g. adding a spurious turn to the route).

5.5.1 Inferring procedures implicit in instructions

Implicit procedures are inferred using knowledge and reasoning about both language and the task, here large-scale spatial navigation. For instance, reading “Go down the hall to the chair,” MARCO interprets the phrase structure as *along* and *until* parameters of a Travel procedure. Using spatial knowledge and the Travel^p action model, MARCO infers the conditions of the Travel^p procedure: (**Pre**) the path should be immediately in front and the chair should be in the front in the distance and (**Post**) the chair will be local to the agent.²

Both Grice’s conversational maxims (1975) and Relevance Theory (Sperber and Wilson, 2004) are linguistic theories of discourse – how sentences are strung together to form broader meaning. Each theory assumes that a cooperative speaker conveys meaning by crafting the discourse to clearly and concisely carry across the necessary concepts.

We make the same broad assumptions as Grice: the director is cooperative with the follower, is descriptive enough to guide the follower (Maxim of Quantity), is generally accurate (Maxim of Quantity), is relevant (Maxim of Relation), and is understandable (Maxim of Manner). However, some of these maxims must be relaxed to handle natural instructions. Because of individual variation and varying motivation, some subjects are too curt, others too prolix.

The follower must be able to ignore extraneous information, but also to fill in implicit procedures. Though directors are generally accurate, they are fallible, especially in remembering turn directions and travel distances (MacMahon, 2005). The follower must be able to account for discrepancies between the instructions and the environment. For instance, the follower should not go forward when the last command has it facing a wall, not a path. Finally, instructions may be ambiguous and under-specified *a priori*, but the spatial context and situated spatial reasoning will usually resolve the ambiguity.

²Note that in earlier work, these were referred to as “implicit actions.” However, they correspond to closed-loop procedures on indeterminate length, not to a single causal action or fixed sequence of causal actions.

5.5.2 Recognizing syntactic, semantic, pragmatic, and exploratory cues

Instructions contain cues to implicit procedures across all of levels of linguistic processing. Syntax, semantics, and discourse pragmatics are all needed for natural language understanding. Here, we describe recognizing the cues of implicit procedures in instructions that are primarily encoded in each level. It is difficult, and probably unnecessary, to draw a clean line demarcating the roles of syntax, semantics, and pragmatics in natural language understanding. However, the methods we present here are principled and should generalize to instruction understanding in different domains and other applications of natural language understanding.

Syntactic Cues for Implicit Procedures

Syntactic cues are syntactic constructions that mark a condition for a procedure which may be achieved by another procedure. For instance, a locative phrase, such as “at the corner,” may require a *Travel^p* procedure to reach the location. Similarly, a phrase describing a pose, e.g. “facing the hallway,” may require a *Turn^p* before proceeding with the procedure in the main clause. The instruction modeler recognizes some linguistic conditional clauses as syntactic cues, e.g., “when,” “at,” and “with,” as well as purpose clauses, (e.g. “Turn so that the chair is in front.”). These conditionals are modeled as possibly requiring an embedded procedures to achieve.

Syntactic cues are domain and verb independent. They are marked by a small set of grammatical constructions. Syntactic cues can be evaluated independently of the verb being modified: for instance, in the construction “at *X*, do *Y^p*,” the agent can take actions to achieve *X* even without understanding *Y^p*. For route instructions, *X* is a location, which may require a *Travel^p* to achieve. In other domains, *X* might be some other point-like state, such as a temperature in a cooking recipe.

Though syntactic cues mark implicit procedures independently of spatial context and reasoning, the application of the cue to navigation is still a pragmatic *contextual*

implication (Sperber and Wilson, 1986, 2004). That is, the agent can recognize that a procedure may be needed through syntax, but determining which procedure is pragmatic reasoning. Consider the sentences “Go down the hall toward the chair. At the blue hall, turn right.” Is the chair before, at, or after the blue hall during the first travel? Alternatively, the environment might constrain where the follower can turn when traveling toward the chair, and the blue hall may only be visible after the turn. In large-scale space, the follower can only determine which procedures are needed by moving through and perceiving the environment.

Semantic Cues for Implicit Procedures

Semantic cues may imply unstated procedures in two different ways. First, the semantic frame may encode a procedure (Bindiganavale et al., 2000; Chang et al., 2002; Tellex and Roy, 2006). For instance, “Take the third left,” implies a *Travel^p* to the third intersection with a possible left turn, before the *turn^a* to the left. Second, a phrase may have the primary role of controlling the main procedure, but a secondary, implicit role of giving a precondition to satisfy. For example, in “Walk down the hall until the chair,” the *until* phrase not only is the termination condition of the *Travel^p* procedure, but also implies a *Turn^p* to face the chair before beginning the *Travel^p*. At the least, a chair should be possibly reachable, if not visible, in the direction faced – not, for instance, a short dead end path without a chair.

Semantic cues are domain dependent, depending on the verb. Consider two similarly constructed instructions: “Turn left past the chair” and “Walk forward past the chair.” In the *Turn^p* frame, the *past* phrase marks a precondition location; chair must be *behind* the follower before the *Turn^p*. In the *Travel^p* frame, the *past* phrase implies a series of conditions; before the *Travel^p*, the chair is *in front* of the follower, *by* the follower during the *Travel^p*, and only *behind* the follower at the completion of the *Travel^p*. Contrast this to the verb-independent implications of syntactic cues, such as *at*, explained

in the last section.

Pragmatic Cues for Implicit Procedures

Pragmatic cues become important when the conventions or biases of the domain require interpreting some utterances differently than their surface meanings. Pragmatics can either be at the utterance or discourse level. At the level of a single utterance, idioms and conventions are interpreted differently than their syntactic and semantic models. For example, a declarative sentence with a pronoun and a locative expression may be interpreted as an imperative sentence, especially at the end of the instructions. e.g. “It’s at the end of the hall,” may mean “Go to the end of the hall.” Domain-specific discourse pragmatics can come into play to fill in procedures that are conventionally skipped in the instruction text, but are not implicit in syntactic or pragmatic preconditions. For instance, “Turn left and then left again,” usually implies a *Travel^p* procedure between the two *Turn^p*s.

Additionally, pragmatic cues may trigger the combination of information from separate utterances into one procedure specification, or the interpretation of one utterance differently depending on the context of its prior and next utterances, if any. Early implementations of MARCO did not combine information across separate clauses, other than pronoun resolution (MacMahon and Stankiewicz, 2006; MacMahon et al., 2006), nor did other similar work in procedural route instruction following (Bugmann et al., 2004; Simmons et al., 2003).

Each procedural specification has an index value and a pointer to the full instruction plan. MARCO has various heuristics to integrate information across utterances or based on the position of the utterance in the instruction sequence. One implemented cross utterance pragmatic heuristic fills in a missing termination condition for a *Travel^p* procedure (the *until* condition) from the linguistic context of the upcoming locative and descriptive phrases. Another heuristic inserted a *Travel^p* forward when the final procedure is an explicit *Turn^p*.

Yet another inserts a `Travelp` to the next match, when the until condition of a `Travelp` is met immediately after an explicit `Turnp`, e.g. “At the end of the hall, turn left and then turn right at the end of that hall.”³

Exploratory Implicit Procedures

Exploratory procedures are when the follower acts to gain knowledge. These are not purely what have been called “knowledge-producing actions” (Scherl and Levesque, 2003), because they have side effects in the world. The above implicit procedure inferences can all be implemented in the instruction modeler or the executor. Though the executor primarily performs the procedures stated explicitly or implicitly in the route instructions, the executor also plans sequences to gain information and to achieve pre- and post-conditions of procedures. Exploratory procedures may be necessary to determine where a reference object is: e.g. in “Go away from the chair,” the follower may `Turnp` to locate the chair. If the pre- and post-conditions of procedures are not met, the executor executes a contingent plan to achieve them. The procedures the follower takes depend on both the route instruction text and the text’s correspondence to the environment.

For some exploratory procedures, such as “Go toward the longer end of the hall” or the above “Go away from the chair,” the follower may start in a pose satisfying the condition. However, without the knowledge of what is behind, the follower does not have certainty that the condition is achieved. Thus the exploratory `Turnp` may undo an achieved condition in order to gain information about the world, before re-achieving the condition. Of course, if the follower already has that knowledge, no implicit procedures are necessary – the follower can proceed with the `Travel` in these examples.

³Note that the differences in determiners are not enough to differentiate between the halls, as `the` and `that` can refer to the same hallway, as in `Face the blue hall and go to the end of that hall`.

5.5.3 Executing an Example from the Route Instruction Corpus

Consider the sentence “Take the blue path to the chair.” Figure 5.2 shows how this instruction is applied to navigate given different maps and starting poses. Figure 5.2(a) shows the default assumption, that the previous instruction elements have moved the follower into position. If a blue path is immediately in front of the agent, it will execute the explicit *Travel^p* procedure. In Figure 5.2(b), the blue path is visible immediately to one side, so it will *Turn^p* to meet the precondition of *Travel^p* along a path, though this procedure is not stated in the instruction (a semantic cue). In Figure 5.2(c), the blue path is visible to both sides, but the follower does not know which way the chair is. The follower must make an exploratory *Turn^p* to look down the blue hall in one direction, then if it does not see the chair, *Turn^p* around to face the chair.

If the follower does not see a blue path immediately in front of it, but does see one off in the distance (Figure 5.2(d,e)), it will *Travel^p* to the distant path, then *Turn^p* onto it before proceeding. Figure 5.2(f) shows the agent making an exploratory *Turn^p* to find the blue hall, a *Travel^p* to reach it, another exploratory *Turn^p* to find the chair, and only then the explicit *Travel^p* command. If it does not see a blue path from any pose at its current location, it will move through the environment until it finds a match, with a small, but increasing, chance of giving up. This search behavior improves performance on poor instructions, while not significantly reducing the success rate of highly-rated instructions (MacMahon and Stankiewicz, 2006).

5.5.4 Other work on understanding implicit procedures

Most other work on implicit procedures in route instructions has focused on the semantic cues for implicit procedures, especially understanding how to achieve the preconditions procedures. Tellex and Roy implement spatial routines that achieve the preconditions of commands within the perceptual surround of a robot (Tellex and Roy, 2006), for instance taking the spatial context into account to move to the next opening before executing a

“Go right” command. Both Tellex and Roy (2006) and Simpson (2005) survey robots and smart wheelchairs that take spatial context into account when following commands. However, this and other work only accounts for single instructions and does not test when the implicit procedures are necessary in the linguistic context of a stream of instructions.

Bugmann et al. (2004) implemented a robotic system capable of following programs of functional primitives from a corpus of 144 route instructions (See Section 2.4.4). They modeled 15 functional primitives each taking an optional parameter list. They split similar procedures into several procedures, where we would just have one procedure. For instance, they model `go_untilp`, `exit_roundaboutp`, `follow_road_untilp`, and `take_roadp`, all of which would be modeled with our `Travelp` procedure with various keyword parameters.

Bugmann and colleagues handled some of the same implicit procedure cues as we do, although they do not break down how much each contributes to performance. They handle some semantic and syntactic cues, giving examples of “Turn left” meaning continue forward until a left turn is possible, then turn; and “At the second intersection, turn left,” meaning travel to the second intersection, then turn left. However, the examples they give as shortcomings of their agent show inability to recognize discourse pragmatics and exploratory cues: not noticing a dead-end on a wrongly described turn and not integrating information between sentences in “Pass first intersection. At the second intersection turn left.”.

5.6 Robot Controller

The *Robot Controller* module executes the low-level `turna`, `travela`, `verifya`, and `declare-goala` actions. Robot controllers present a common interface to the executor, with domain-dependent implementations. The controller moves the agent for `turna`s and `travela`s and can `verifya` if a view description matches an observation.

The robot controller acts at the level of the Control tier of the 3T intelligent architecture. The robot controller in GRACE is written as commands and monitors in TDL.

The robot controller is an interface, presenting the executor discrete Causal actions such as *turn^a*, *travel^a*, and *verify^a*, hiding the Control level of the SSH ontology (continuous sensorimotor experience and control laws). The *turn^a* and *travel^a* actions are translated into movement; The *verify^a* actions into perception, and the *declare-goal^a* action to a signal to end the way-finding, possibly stating that the robot is at the goal or by taking a special action in a simulation.

The actions the robot controller takes are dependent not only on the instructions, but also models of the world and the robot's capabilities. For instance, consider the instruction "Move to the blue hall." The executor executes the robot's *travel^a* action and then the robot's *verify^a* action. A controller for a robot with peripheral vision would execute the *verify^a* by analyzing the periphery of the view while facing along the travel direction. A controller for a robot without peripheral vision would *turn^a* to face each hallway, *verify^a* if it is blue, otherwise *turn^a* back, all as part of the *verify^a* action.

5.7 View Description Matcher

The *view description matcher* checks the symbolic view descriptions against sensory observations. The view description matcher treats the view description as structured constraints that the observation stream must meet. This defers handling many forms of ambiguity until the environment can provide some disambiguating context. The view description models the world by recognizable entities, such as objects or spatial configurations, their attributes, such as size or color, and the relations between them, including spatial, compositional, and logical relationships.

The view description matcher is the code that grounds the symbolic relational representation in the sensory experience. This may involve resolving ambiguities among possible senses of a phrase. For instance, "Face the brick path" may mean "Face along the brick path immediately in front of you" or "Turn towards the brick path that is visible in the distance." The view description verification code can check the observation for each of

these meanings.

For instance, given the instruction “Turn to face the blue path,” the view description would be *Path side:Front, appear:blue* . The view distance is unspecified, because it is unconstrained: The blue path may run forward from the agent $P(\text{distance:}'0', \text{side:Front})$ or may be visible crossing this path in the distance $P(\text{distance:}'1:', \text{side:Sides})$. MARCO checks for both cases while turning.

The view description matcher performs unification (Russell and Norvig, 1995) between the under-specified constraints modeled in the view description and a perceptual model of the follower’s place in the environment. First, the view description is a partial description of the observations the follower should encounter at that point in executing the instructions, only mentioning some aspects of some of the landmarks the follower will encounter. Second, the descriptions of the attributes are also under-specified: a path “in front” may lead forward from the agent’s location or merely be visible in the distance. Even matching concrete referring phrases to percepts through object recognition is only a constraint: in the corpus for this dissertation, directors use “chair” to refer to three perceptually different pieces of furniture: a stool, a bench, and a dining chair, even at times when more than one is visible. Other noun phrases are even more vague, e.g. “piece of furniture” or “something.”

There is a clean interface layer between the executor and the view description matcher: the executor mostly treats the view descriptions as an opaque data type, formed by the instruction modeler and verified by the view description matcher. The executor only performs simple transformations on some parameters of the view descriptions, for instance, projecting a desired local view $P(\text{distance:}'0', \text{side:At})$ into the distance $P(\text{distance:}'1:', \text{side:Front})$. This is a form of *perspective taking*, a fundamental spatial skill in people (Schober, 1993; Trafton et al., 2005).

The view description matcher will use whatever perceptual abilities the robot has available. On a hardware robot, the concept of an intersection can be linked to the code

that segments intersections in the laser scan and classifies the local path topology (e.g. as a dead end, “T”, or corner intersection (Kuipers et al., 2004)).

With the simulation in this paper, MARCO cannot directly observe intersection type, but must model it through the relative positions of the observed paths (see Figure ??). In these instructions, “the corner” usually refers to a “L” intersection: the intersection of two paths, each terminating at the corner, so the view description matcher procedure for corner looks for a path configuration (local topological map) that meets these constraints, local to the follower or off in the distance. Other constraints, e.g. “the corner of the red and blue paths with a chair,” are applied to unify the constraints with the observation and perceptual model.

The view description matcher encapsulates the following capabilities: object recognition (Modayil and Kuipers, 2006), local topology recognition and local metrical mapping (Kuipers et al., 2004), small-scale spatial referring phrase resolution (Regier and Carlson, 2001; Skubic et al., 2004b), perspective taking (Schober, 1993; Traflet et al., 2005), and visual search.

5.8 Modeling Route Instructions in the HSSH Ontology

The *Spatial Semantic Hierarchy (SSH)* (Kuipers, 2000) and its extension the *Hybrid Spatial Semantic Hierarchy (HSSH)* (Kuipers et al., 2004; Modayil et al., 2004) are layered ontologies of space. The representations of the HSSH capture reasoning and acting in small- and large-scale spaces. Small-scale spaces are areas within the sensory surround of an agent, such as a room or an open field. The small-scale space for an agent is the area it can map by turning around and moving within its sensory horizon. Large-scale spaces extend beyond the sensory horizon, whether because of opaque obstacles, such as in a building, or the natural limits of perception, such as in the open desert, plains, or ocean, where the agent can see to the horizon.

Route instructions describe causal and topological structures annotated with

metrical and rich view (object and landmark) information. A *view* abstracts the sensory image to a symbolic representation. In the SSH, the *causal* level discretizes continuous control motions into reliable discrete actions. At the causal level, motions are abstracted to either *turn* or *travel* actions. A *turn*^a action changes orientation within a place, while a *travel*^a moves the agent from one place to another.

In the HSSH, the four levels are rethought as the cross-product of interpreting space locally or globally and metrically or topologically. The levels are the *local metrical*, *local topological*, *global topological*, and *global metrical* levels (Kuipers et al., 2004). In each, representations and actions in the higher (later) levels build on representations and actions at the lower levels. For instance, at the local metrical level, the agent maps obstacles and plans safe motion in the continuous world. At the local topological, the agent recognizes, reasons about, and moves between symbolic intersections.

The *topological* level of the Spatial Semantic Hierarchy represents the environment as *places*, *paths*, and *regions* and the topological relations of *connectivity*, *path order*, *boundary relations*, and *regional containment* (Remolina and Kuipers, 2004).

Parts of route instructions can be represented by the SSH causal and topological ontologies, with the actions annotated with metrical and view attributes. Route instructions include both causal actions (“Walk forward three steps.”) and topological actions (“Face along the blue path and follow it to the the brick hall.”).

MARCO’s executor module interprets reactive procedures in the spatial context to execute causal actions, although the robot may recognize and reason about topological entities such as paths and intersections. Each *turn*^a and *travel*^a moves the follower to the next pose and each *verify*^a compares the view against the view description. However, moving up the ontology, an executor can reason about the spatial layout of the route. Explicitly reasoning at the topological level can help handle ambiguity in the language, interpretation, observation, execution, and map learning.

The executor reasons at all four levels of the HSSH. In the SSH, these are the

control, causal, topological, and metrical levels. (Kuipers, 2000).

5.8.1 Relation to the Spatial Semantic Hierarchy

The Spatial Semantic Hierarchy is a rich, but complex model for reasoning about space. Is the SSH excessive or just complex enough for the needs of following route instructions?

First, the Spatial Semantic Hierarchy is a well-developed, theoretically grounded cognitive spatial model. The SSH is supported by having working implementations on mobile robots (Kuipers et al., 2004; Kuipers and Byun, 1991), and by modeling human navigation and spatial learning performance (Kuipers et al., 2003).

The SSH is a lattice of representations that can model states of partial spatial knowledge, including partial maps as the environment is learned. For instance, the SSH can easily model knowing only some routes instead of the overall spatial network; knowing that two paths meet a corner, but not the relative turn direction between the two; or knowing some path segment distances but not others.

Route instructions must select certain aspects of the route to describe, so by nature under-specify the route. Instruction texts do not often mention the absolute location of everything in the environment. The SSH can model the director eliding a turn direction, occasionally mentioning a distance, and selectively noting landmarks and other perceptual features.

The SSH has several advantages over global metrical maps, such as occupancy grids. With a known global metrical map, a robot can follow a set of route instructions, although finding the route in the metrical map will require processing on a hybrid, topological representation. However, since the global metrical map is a precise representation of what has been observed, it is difficult to reason about unseen, unknown places. To navigate with a global occupancy grid alone, the agent must know the exact dimensions of the hallways and rooms, which are never described in natural language route instructions. Generating route instructions using a metrical map is possible, but

using a representation that more closely follows people’s cognitive maps should produce instructions which are more robust and easy to follow.

Moreover, the SSH first and fundamentally models the environment in terms of its *navigational affordances* — its paths, places, and intersection structure. This is exactly the sort of information described in route instructions, the navigational semantics of places on the route. To follow instructions, having a high-level, but semantically annotated map is necessary, where a metrically precise map of occupied space will fail. The agent must be able to recognize *places*, *paths*, and *intersection type*, not just the presence or absence of obstacles.

The multi-tiered SSH ontology allows the follower to recognize and act at the different levels people do. See Section 6.7 for an experiment showing the necessity and impact of the different HSSH levels for following route instructions.

5.8.2 Modeling route instructions by topological maps

Route instructions fundamentally describe a route, a topological trace through a large-scale space. Since the route is composed of topological path segments, one approach would be to model the instructions declaratively, as an under-specified topological map of the environment being traversed. Then, the follower could navigate using traditional methods for way-finding on a known map with uncertain movement (Cassandra et al., 1994; Fox et al., 1999; Kaelbling et al., 1998; Koenig and Simmons, 1996, 1998; Simmons and Koenig, 1995; Theodorou et al., 2004).

Resolving linguistic and perceptual ambiguity while following route instructions is analogous to resolving perceptual aliasing. Route instructions do not completely specify the route, leaving spatial ambiguity. For instance, a turn direction may be unspecified, leaving topological ambiguity.

Procedural specifications model the route instructions as a list of imperative procedures. The follower concentrates on inferring what the director intended the follower

to do. However, route instructions also state or imply a spatial route layout. Another approach to following route instructions is to extract the implied map of the route from the route instructions. The follower infers what the director intended the follower to know about the route map.

Deriving the possible topological route maps from the route instructions is an attractive idea. Perhaps the hybrid topological SLAM algorithm in the HSSH that handles spatial and perceptual ambiguity (Beeson et al., 2007; Kuipers et al., 2004; Modayil et al., 2004) can handle the ambiguous maps derived from under-specified or linguistically ambiguous route instruction. That is, can the partial, ambiguous map of the environment derived from language understanding and the partial, ambiguous map learned from exploration be represented and reasoned about in the same way by the same cognitive processes?

Tractability of topological SLAM for route instruction following

After careful examination, performing route instruction following by SLAM on a set of derived maps is intractable. Route instructions provide primary local guidance information, without providing any global topological information. Moreover, the type of information that route instructions provide conflicts with the axioms of topological SLAM (Remolina and Kuipers, 2004). Instructions rarely mention all places between two turns. The topological reasoner, on the other hand, constrains search by assuming it knows all places along a traveled path segment. Therefore, a described travel procedure in route instructions cannot be treated the same as a travel trace along the described path segment to the topological mapper.

Moreover, the information in the view descriptions is under-specified, leading to more perceptual aliasing of places. Finally, some of the information may be inaccurate. In fact, the least accurate information in route instructions is the causal information of turn directions and distance counts, which the topological mapper fundamentally relies upon

(MacMahon, 2005).

Though route instructions and sketch maps appear to be derived from the same underlying topological cognitive maps (Tversky and Lee, 1999), route instructions are less well specified. The sketch map provides a topological map of the route with taken decision points on the route explicitly connected by travel arcs. In other words, the number of turn and travel procedures, and the places they begin and terminate, are explicit in complete route sketch maps. In verbal route instructions, some of these travel arcs and turn places may be implicit, so each set of route instructions may correspond to many topological route maps, and therefore many sketch maps.

Even worse would be applying Markov localization in a straight-forward way. Markov models rely on having a complete description of the task and environment. For localization, this means the likelihood of moving to each pose from another given an action, and of the likelihood of seeing each observation from each pose. Where route instructions under-specify the possible observations (that is, on almost all poses), we must account for this under-specification.

The most direct approach treats the Markov observation set as the set of all possible feature vectors. Yet in the general case, the set of feature vectors for route instructions is unlimited, as the director may characterize various attributes of the environment at various levels of detail, approximation, and vagueness. However, Markov models require a finite set of observations, so this cannot work.

A second approach is to treat the observation as a feature vector representing whether the perception matches the view description for all poses in the environment. The catch is that view descriptions are not mutually exclusive. Where one describes a chair in front, another will describe a path off to the right two intersections away. A priori, the two view descriptions may be match the same pose, even if the poses are widely separated. However, other perceptions will match only one of these view descriptions. So the Markov observation must capture the set of view descriptions for poses that the perception matches.

This implies that for n poses with n different view descriptions, there will be 2^n possible observations – the power set of possible matches.

The observation space explodes to the power set of combinations because different view descriptions will describe orthogonal aspects of the view, e.g. `the blue hall` and `a chair in front`. For any given percept, one, both, or neither of these descriptions could match. Describing a third attribute multiplies each combination by another true or false match.

The exponential size of the observation space is bad enough on its own. Worse is that to solve a POMDP is doubly exponential in the size of the observation space (Kaelbling et al., 1998). An observation space that scale exponentially with the size of the state space – number of poses mentioned – adds a third level of exponentiation, so that even solving even short routes is intractable. Worse, this has the paradoxical result that the more detail the director provides (especially about poses between or past turn poses) the slower solving the POMDP becomes. As the director gives more information, each new bit of information multiplies the difficulty of the problem that the Markov follower must resolve, because there are more places the agent must rule out.

Though the general case of performing topological SLAM on the complete route map is intractable, using Markov decision processes locally is an interesting avenue of future research. Once the agent has perceived the scene and modeled the local surround, the medium-scale space of the surrounding hallways are much better specified than from the instructions alone. The follower need not enumerate all the worlds possible from the instructions alone over the entire route. Instead, the follower can use the POMDP to make local decisions on a perceptually filled-in model of the local surround. This may still prove to be a practical way to handle locally resolvable ambiguities.

5.9 Extension to handle other sorts of ambiguity

To date in this architecture, we have applied deferred resolution of ambiguity for matching referring phrases to observations and to modeling the instruction text as an under-specified plan of action. In the first case, the ambiguities are in lexical and noun-phrase semantics and at the perceptual levels. In the second case, in the semantics and pragmatics of verbs and sentences and at the procedural level.

These same principles for resolving ambiguity through interaction with the world can be applied to other processes in instruction following, such as selecting the best syntactic parse and pronoun resolution. Parsing is the problem of picking the most probable syntactic structure given a series of words. Pronoun resolution is the process of selecting the most probable link between pronouns and other entities in the discourse or shared context, including *anaphora* – matching a pronoun to an antecedent phrase, *cataphora* – matching to a following phrases, and *exophora* – matching to an entity in the world that is otherwise referenced in the discourse (Bos, 2004; Byron et al., 2005; Kamp and Reyle, 1993).

For the syntactic ambiguities of parsing, using a re-rank parser such as (Charniak and Johnson, 2005), the executive layer can re-rank the list of most likely parses based either on the higher level semantic constraints within the utterance or on pragmatic constraints from the surrounding linguistic and task context. For instance, in “Move towards the chair on the blue floor,” the phrase “on the blue floor” may specify the location for the chair, the path for travel, or both. If the mostly like initial parse is that the phrase is modifying the chair, but the only visible chair is past the end of the blue path, then the system should re-rank the parse list. Similar ideas are seen in (Fleischman and Hovy, 2006; Fleischman and Roy, 2005; Skantze, 2005).

Deferred resolution can be applied in a similar way to handle the semantic ambiguities of pronoun resolution. For instance, in “You should be facing flowered carpet and an easel, move to it.” does it refer to the carpet or easel? This idea can build upon similarly motivated work applying semantic filtering and visual

attention to reference resolution (Byron, 2002; Byron et al., 2005; Knees, 2002).

“Take the blue path to the chair.”

$\text{Travel}^p(\text{along:Path}(\text{appear:Blue, side:Front}), \text{until:Chair}(\text{distance:}^{\circ}0^{\circ}, \text{side:At}))$

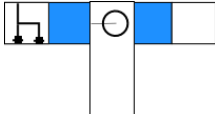
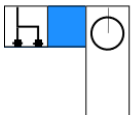
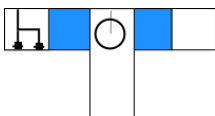
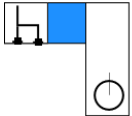
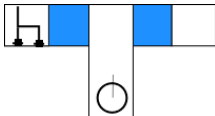
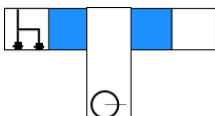
Map, Robot Pose	Implicit Actions	Worst-Case Actions Taken
a 		Travel
b 	Precond Turn	Turn Travel
c 	Explore Turn Precond Turn	Turn 2 Turns Travel
d 	Precond Travel Precond Turn	Travel Turn Travel
e 	Precond Travel Explore Turn Precond Turn	Travel Turn 2 Turns Travel
f 	Explore Turn Precond Travel Explore Turn Precond Turn	Turn Travel Turn 2 Turns Travel

Figure 5.2: How interpreting an utterance depends on the follower’s knowledge of its pose in the environment. The circle represents the follower, with the line in front. The follower sees hallways to its side, but not down the side hallway.

Chapter 6

MARCO Route Instruction Corpus Experiments

The route instruction language and task corpus captures how people describe large-scale space, in particular, the task of spatial way-finding through an unknown environment. The corpus measures the individual differences in how people describe a spatial route, including mistakes and omitted information. The action traces and *post-hoc* ratings of the human followers show how well people can apply these faulty, under-specified instructions to accomplish the concrete task of navigation.

This corpus provides empirical data on the spatial, perceptual, linguistic, and executive skills needed to follow the imperfect route instructions that people actually give. By building a system that can approximate human performance, we can show what is sufficient to follow natural language route instructions. By ablating (selectively turning off) these various skills, we can measure how often particular skills are necessary to follow natural language route instructions.

This chapter will first describe the general procedure for running MARCO through the instruction corpus. Next, we present a series of ablation comparison experiments, examining the roles of implicit procedure inference, landmark recognition, and spatial

reasoning skills in following natural language route instructions.

6.1 MARCO Followers Model the Text and Navigate

In general, MARCO follows the same basic procedure as described for the human followers in Section 4.7. For each instruction text in the corpus, MARCO is put at the starting place, facing an arbitrary direction. Like the human followers, MARCO does not have any *a priori* map or other knowledge of the environment or route layout. MARCO parses and models the instruction text to navigate to the destination, given first-person views of the virtual environment. When MARCO is finished following the instructions, it must explicitly take an action to terminate navigation, just as the people did. Unlike people however, MARCO does not rate the instructions after following them.

6.1.1 Apparatus

We are interested in measuring how often different skills are necessary to follow natural language route instructions, but, at this point, not in the particular implementation of those skills, especially the perceptual skills. Therefore, MARCO navigates through a symbolic interface to the same environments that people saw as a three-dimensional scene. The correspondence between the visual and symbolic views is described in the next section.

The symbolic navigation environment was coded in Python (Python, 2007). The code implements a discrete motion simulation of the environment by calculating the appropriate view and environmental observation after each action. The environmental representation used is a Markov model, though this is completely hidden to MARCO. Though all actions and observations are deterministic in the current experiment, the action model can handle both non-deterministic actions and observations.

One advantage of this model is speed of simulation. Since the model does not need to render a three-dimensional scene, as the Vizard simulation does, it can simulate a navigation run very quickly. This allows fast regression and ablation testing, running

through the entire corpus in minutes per follower instance. Following 1500 instructions on a physical robot in a real-world environment in a comparable amount of time is impossible.

6.1.2 Stimuli

The symbolic perceptions given to MARCO provide the same information in the same structure as the visual scenes presented to people. Figures 6.1 and 6.2 show two examples of the correspondence between the Vizard scene and the symbolic model view.

In Vizard, the human participant can see bits of any local peripheral hallways and all the way down any straight hallway segment in front, until a wall blocks the view. The symbolic view represents the same information – which entities are visible to the immediate and peripheral front of the embedded way-finding agent. The symbolic view represents the base components of the environments – the object landmarks (Table 4.1) and hallway textures and wall pictures (Table 4.2).

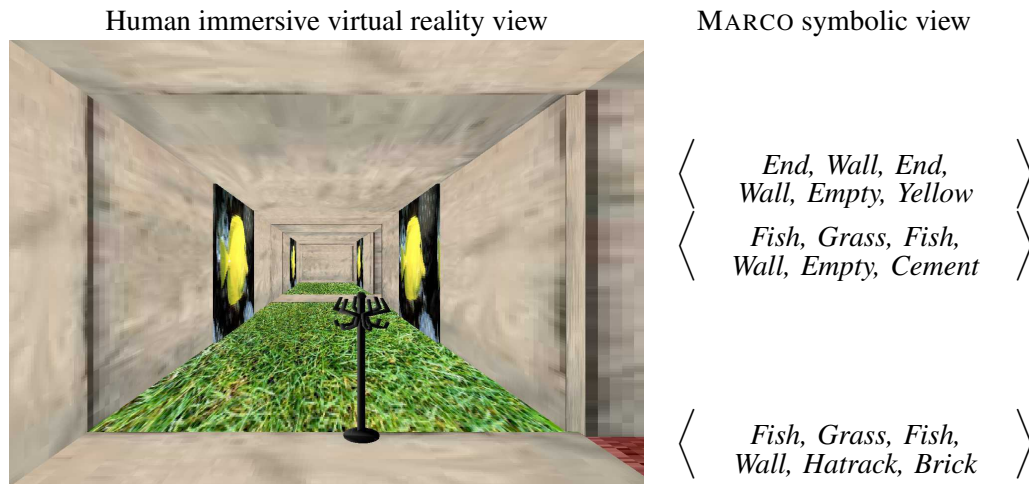


Figure 6.1: Human visual view and MARCO symbolic view of a hallway. The symbols in the tuples correspond to the walls, paths, furniture, and floor textures visible at the same relative distance in the view.

In the current experiments, MARCO receives a symbolic observation consisting of a list of tuples, one tuple per intersection visible in front. Each item in the tuple represents

one of the components of a view of an intersection – the hallways or walls visible to the left, front, and right, any furniture in the intersection, and pictures visible on any corridor walls to the front. The relative ordering within each tuple corresponds to relative position within the visual scene. For instance in Figure 6.1, the immediate intersection has a wall on the left and a brick hallway visible on the right, as does the lower line of the bottom tuple in the symbolic view. The relative ordering of the list of tuples corresponds to relative ordering of the view down the path in front, if any.

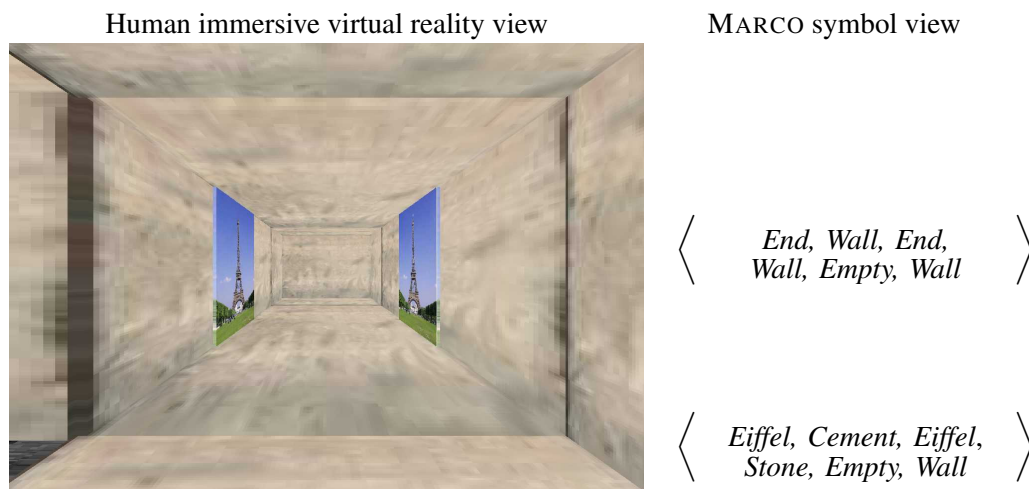


Figure 6.2: Human visual view and MARCO symbolic view of a shorter hallway

Because the list of symbol tuples corresponds to the same entities used to derive the virtual reality visual scene, the visual and symbolic models contain nearly the same navigation information. The only difference is that the symbolic view does not model the position of object landmarks within the small-scale space of the intersection, so the symbolic view is slightly less informative than the visual scene.

Note that the symbolic view represents the base constituents of the environment, but may not correspond to how the directors refer to the environment. For instance, while `chair`, `hallway`, and `Eiffel tower` are all directly represented, other nouns, such as `corner` and `furniture` are not, nor are more complex referring phrases such as `the`

corner of the red and blue hallways with the chair or the longer end of the yellow-tiled hall. The agent has to have some knowledge of the ontology of the environment and how to match a complex referring phrase to a view, compositionally.

This symbolic observation representation is an implementation detail that MARCO does not depend upon in any way. The instruction execution core is separated from the recognition and local perceptual modeling code by an abstraction layer.

6.1.3 Procedure

To follow a set of route instructions, like the human followers, MARCO is placed at the starting position, facing an arbitrary position. It has the route instruction text and attempts to navigate to the destination. In most of the experiments below, MARCO starts with the gold standard parse trees for the instructions, although Section 6.8 examines parsing the raw text using a learned grammar.

MARCO navigates through the environment until the instructions end or until it cannot find a described view. After each trial, MARCO is placed at the next starting position and given a new set of instructions. MARCO builds up local perceptual models of the environment while traveling, for instance, to remember what is behind it. These local perceptual models are the symbolic views in each direction MARCO has seen, shifted as MARCO turns or moves forward. However, it does not build global models of the route or environment and does not remember its local models between trials. MARCO forgets the local perceptual map of one hallway as soon as it moves off of it.

6.1.4 Evaluation

We compare the empirical performance of MARCO to the performance of human followers on the instructions that people followed and rated. For the purposes of this comparison, a correct route navigation terminates at the requested destination. In some cases, the instructions contain errors or were written with the wrong destination in mind. For about

11.8% (179 of 1522 route instructions) of the corpus, a majority of the human followers terminate at a common place other than the requested destination or the starting location. (MARCO terminates at this same non-target, non-start mode place in 81 of the 179 cases on at least one run.) Though followers that also reach this common point may be regarded as correctly following instructions that lead to a different destination, for the purposes of this study, they are incorrect. Since the human followers are held to the same standard, the comparisons are valid.

We will use the performance of people following the instructions as the gold-standard of how well the instructions can be followed. This controls for both the quality of the instructions and the difficulty of the routes. Where people do poorly, we can expect the task is more difficult, so the performance of multiple followers on an instruction and over a corpus of instructions, gives a benchmark for evaluating the performance of the system.

Likewise, the performance of the full MARCO system can be used as a benchmark for the performance of ablated or differently configured versions of the system. This provides two benefits. First, these experiments measure how often a change makes a measurable difference in the performance of the system, the *impact* of that change. Second, these measurements give insight into how people generate, interpret, and follow route instructions, especially where the changes in performance with an ablated system match a drop in performance for some of our human followers.

Let us call the performance difference, in percent from the baseline MARCO performance, the *impact*. For instance, if the baseline system successfully follows 66% of the route instruction corpus and an ablated version successfully follows 33% of the corpus, the *impact* of that feature was 50% – there is a 50% reduction in measured performance. This indicates how often the ablated feature was necessary to apply the instruction to accomplish the task, normalized by the baseline performance.

The tables in this chapter present the results from the ablation tables using tables with common conventions, e.g. Table 6.3. Each row represents the mean performance over

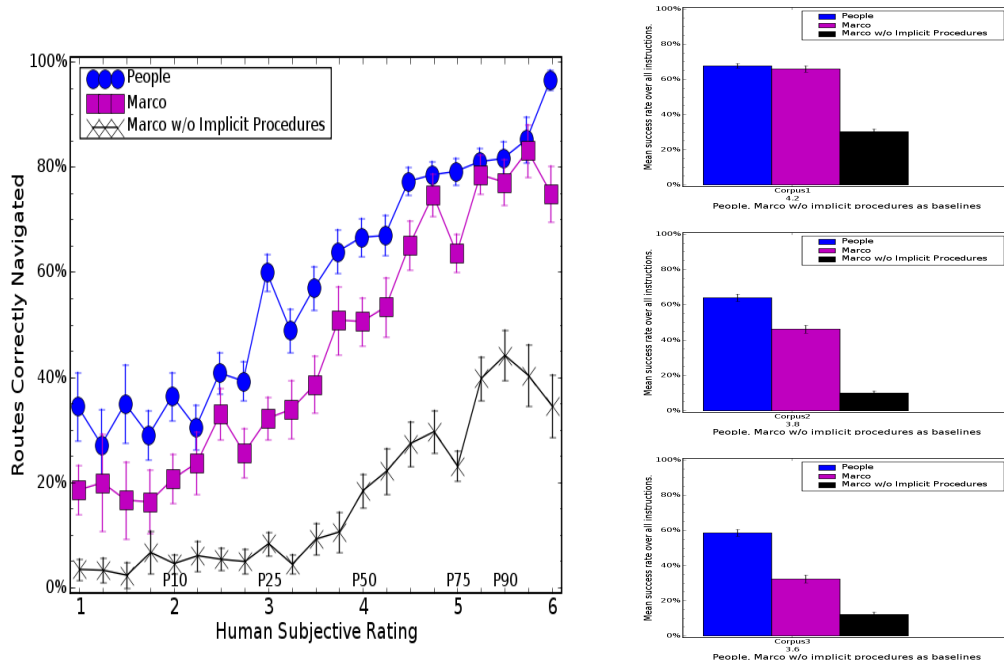


Figure 6.3: Human and full MARCO success rates, with standard error bars, versus *post hoc* human subjective instruction rating. The rating of 1 indicates extremely poor instructions, while 6 is excellent. Success rate is how often, on average, the human followers (circles) and MARCO (squares) finished navigating at the intended position for all the instruction texts with mean human rating of $r \pm 0.125$. Graph on the left shows all data, graphs on right show per corpus. The same trends hold in all three corpora. Data as of May 31, 2007.

multiple runs for a group of human followers or a configuration of MARCO. The columns show the results over instructions with mean *post-hoc* rating within the listed range. Statistical significance on a one-sided T-Test is indicated by the annotations: significance at a 5% level is marked by *, 1% by †, and 0.1% by ‡. These tests and conventions are used throughout this chapter.

Quality Range	1.0–2.5	2.5–3.5	3.5–5.0	5.0–6.0	All	Impt
People	35% [‡]	51% [‡]	73% [‡]	85% [‡]	64% [‡]	-20%
Marco	22%	33%	60%	79%*	52%	0%
Marco w/o Implicit Procedures	5% [‡]	7% [‡]	22% [‡]	40% [‡]	20% [‡]	62%

Table 6.1: Comparing performance of MARCO to people and to MARCO following only explicitly commanded procedures.

6.2 Full MARCO Performance

Figure 6.3 shows the evaluation results comparing people, the full MARCO and the naïve approach of executing only the explicitly commanded procedures. Human participants were able to successfully find and identify the desired destination with an overall mean success rate of 64.3% ($SEM = 0.9\%^1$) of 1522 instructions in the three environments. This was fairly consistent across the corpora from the three experiments: Corpus 1 ($M = 67.7\%$, $SEM = 1.2\%$), $N=691$; Corpus 2 ($M = 64.0\%$, $SEM = 1.9\%$), $N=432$; Corpus 3 ($M = 58.6\%$, $SEM = 2.0\%$), $N=399$. For people, the results are the mean over runs from multiple participants following each instruction set, each beginning at the start location facing a random direction.

With full procedure inference, MARCO successfully followed 51.5% ($SEM = 1.2\%$) of all 1522 route instruction texts followed by people. Further, MARCO increases in performance as the human instruction rating increases and as human performance increases. For the MARCO cases, the presented results are the mean over four runs, facing each of the four directions at the start. While MARCO does not yet match human performance across route instructions of all qualities, the correlations from MARCO’s mean performance to human mean performance ($R_s = 0.958$) and to human mean ratings ($R_s = 0.970$) are extremely strong.

Over the three corpora, MARCO has significantly different performances. MARCO was developed by examining the first corpus, so the performance nearly matches human

¹SEM is Standard Error of Means.

performance: ($M = 66.0\%$, $SEM = 1.7\%$), $N=691$. Controlling for followed routes, this is difference from human performance is not statistically significant, $t(690) = 1.31, p \leq 0.096$. The second corpus with hand-corrected parses and minor enhancements to MARCO, the performance is almost three-quarters of human performance: ($M = 46.2\%$, $SEM = 2.2\%$), $N=432$, a significant difference from human performance, even controlling for route followed: $t(431) = 16.99, p \leq 0.001$. On the third corpus with hand-corrected parses, but no enhancements to MARCO after the Corpus 2 modifications, MARCO successfully followed ($M = 32.3\%$, $SEM = 2.2\%$), $N=399$, over half of human performance, even with worse instructions, significant at $t(398) = 21.01, p \leq 0.001$.

The performance of MARCO following only the explicitly commanded procedures is shown here as a baseline. By executing only explicitly commanded procedures, the ablated MARCO can only follow 19.8% of the 1522 instructions overall ($SEM = 0.9\%$), and between 10% and 30% of the instructions per corpus. The next two sections break down the impact of ablating the ability to recognize different cues to implicit procedures (Section 6.4) and the impact of inferring different navigation procedures (Section 6.5).

6.3 Comparing Implicit Procedures Inference to Fundamental Explicit Navigation Procedures

To gauge how important inferring implicit procedures is to following natural route instructions, we can compare the impact to the impact of ablating fundamental spatial procedures. The $turn^a$ and $travel^a$ causal actions are what move an agent through a large scale space. By removing the ability to recognize the various ways of explicitly commanding $Turn^p$ and $Travel^p$ procedures, we can compare the impact of these core spatial behaviors to the linguistic behavior of inferring implicit procedures for following spatial route instructions.

Each type of causal action can be controlled by an open-loop or a closed-loop

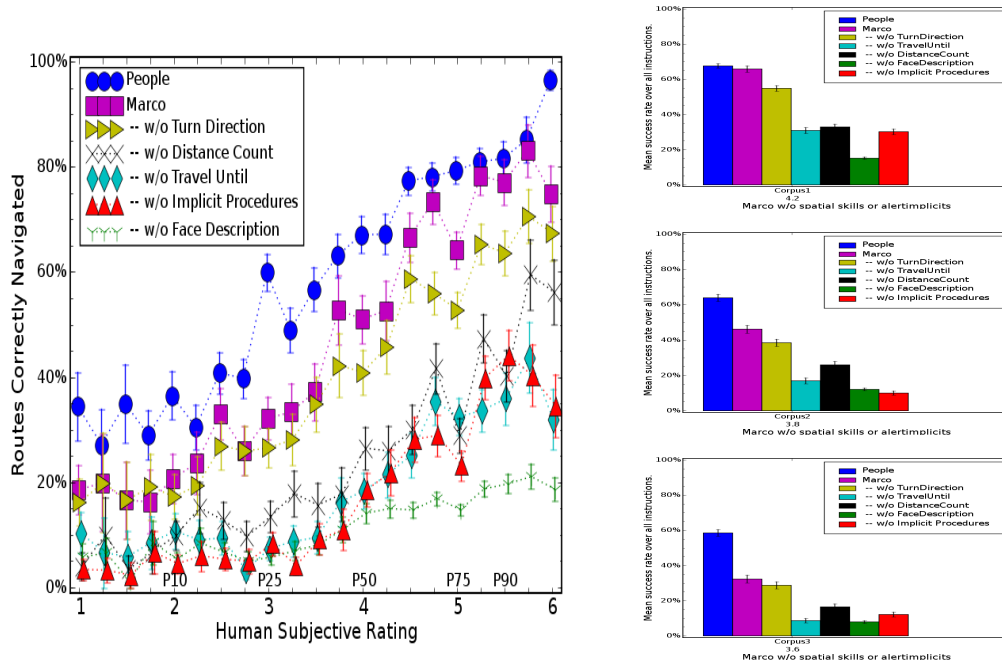


Figure 6.4: Comparing inferring implicit procedures vs. executing types of explicit procedures. The ability to infer implicit procedures is as important as the skills to execute all procedures in the instructions.

procedure. In open-loop procedures, termination is controlled by a relative offset – when an internal condition is met. The open-loop procedures in spatial route instructions are *Turn Direction* for a turn^a and *Travel Distance* for the travel^a. In closed-loop behaviors, the procedure terminates when an external condition is met, when a view description is matched against the local perceptual model. In spatial route instructions, the closed-loop procedures are *Face Description* and *Travel Until* [Description].

Examples of open-loop causal procedures are traveling a distance of a count of intersections (e.g. “Go the third hallway”) and turning to the next possible path (e.g. “Turn left”). Closed-loop causal procedures terminate when a view description is matched by observation. Examples of closed-loop causal procedures are turning to a view (“Face the blue path”) or traveling until a description is met (“Walk forward

Quality Range	1.0–2.5	2.5–3.5	3.5–5.0	5.0–6.0	All	Impt
People	35% [‡]	52% [‡]	73% [‡]	85% [‡]	64% [‡]	-20%
Marco	22%	33%	60%	79%*	52%	0%
– w/o Turn Direction	20%*	29%	50% [‡]	66%*	43% [‡]	16%
– w/o Distance Count	9% [‡]	15% [‡]	29% [‡]	49% [‡]	27% [‡]	48%
– w/o Travel Until	10% [‡]	7% [‡]	26% [‡]	36% [‡]	21% [‡]	59%
– w/o Implicit Procedures	5% [‡]	7% [‡]	22% [‡]	40% [‡]	20% [‡]	62%
– w/o Face Description	6% [‡]	7% [‡]	14% [‡]	20% [‡]	12% [‡]	76%

Table 6.2: Comparing performance of MARCO on inferring implicit procedures and executing types explicitly commanded procedures. Explicitly commanded procedures are executed normally in all cases.

until the end of the hall”).

Just like the ability to infer implicit procedures, we can ablate the ability to recognize these types of explicitly commanded procedures. When ablating one type, the others are in effect, so if the director used a redundant command, the other behavior will still be executed. For instance, given “Take the blue hall three intersections to the chair,” without *Travel Distance*, the follower will navigate to the chair and without *Travel Until*, it will still navigate three intersections. Normally, with both behaviors, it will check both conditions, though favoring the more reliable closed-loop landmark based *Travel Until* when there is a conflict, after moving the described distance. When there is no redundant information, the follower will execute a fall-back default, such as traveling to the next intersection.

Here, we can see that the closed-loop procedures have the largest impact while open-loop procedures have a smaller impact. In the route instruction task, the director does not know which way the follower is facing at the beginning of the trial. This is one reason that *Face Description* had the largest impact. Secondly, closed-loop procedures are both empirically and theoretically more reliable than open-loop procedures. Closed-loop procedures are inherently more reliable because they provide an independent way to check that the procedure was correctly completed. For this reason, the Spatial Semantic Hierarchy

assumes both closed-loop control laws and causal actions are completely reliable (Kuipers, 2000).

Empirically, the descriptions given for landmarks in closed-loop procedures are more reliable than the distance and turn direction parameters given for open-loop procedures in this corpus (MacMahon, 2005). For instance, landmark descriptions were about 10 times more reliable than turn directions (99.5% to 95%) (MacMahon, 2005). Closed-loop procedures allow the follower to recover from his or her own errors while containing descriptions that contain fewer mistakes by the director. (See Section 6.7 for further comparison of open-loop and closed-loop procedures.)

The surprise here is that inferring implicit procedures has a larger impact than the explicitly commanded closed-loop spatial procedures of *Turn Direction* and *Travel Distance* and comparable impact to the fundamental explicitly commanded open-loop procedures of *Travel Until* and *Face Description*. But another way, it is just as important to infer the implicit procedures that the director intended – what the director *meant* – as to execute common invocations of explicitly commanded procedures – what the director *said to do*.

6.4 Implicit Procedures in Route Instructions

Route instructions represent the transfer of knowledge from the director to the follower about specific spatial procedures and environment attributes. Route instructions are useful if the followers reliably reach the intended destination. However, not all the necessary procedures to navigate the route are explicitly asserted as imperative commands. Some necessary procedures are *implicit*, but implied by syntactic, semantic, or pragmatic features of the instructions. The follower must infer the intended procedure sequence from the instruction text. See Section 5.5.2 further description of these cues and how they are implemented in MARCO.

The likelihood of reaching the destination depends on both the instruction set and the skills of the follower for navigation and interpretation. We present results for six types of

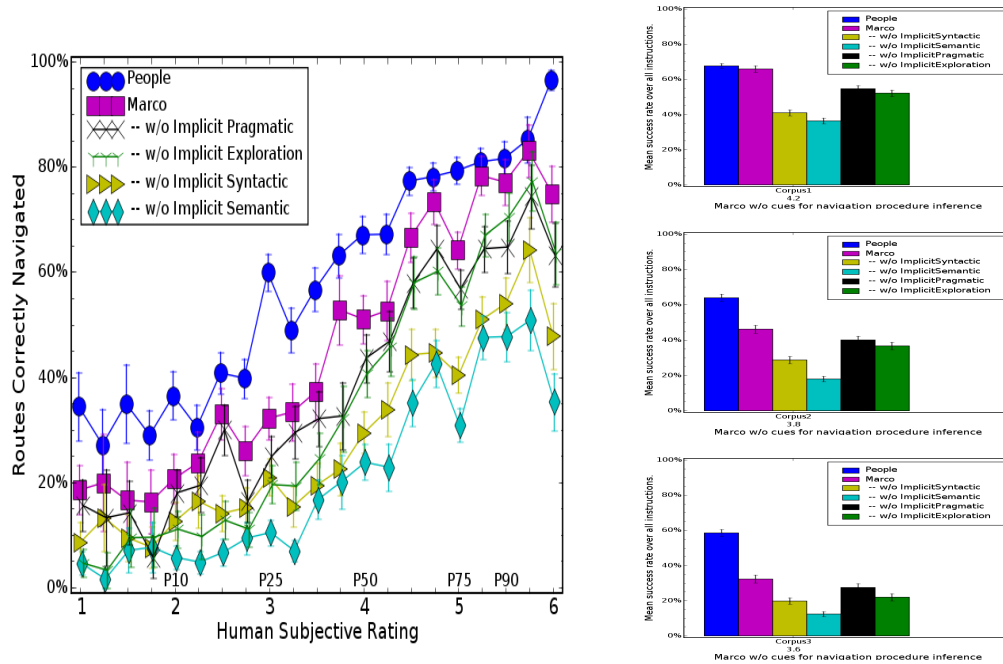


Figure 6.5: Performance for people, MARCO and MARCO without implicit procedure cues, with standard error bars, plotted versus human instruction rating.

followers: (1) human participants, (2) the full MARCO agent, (3) MARCO without *syntactic* cues for implicit procedures, (4) MARCO without *semantic* cues for implicit procedures, and (5) MARCO without *pragmatic* cues for implicit procedure, and (6) MARCO without implicit *exploratory* procedures. Inferred procedures may be unnecessary, depending on the starting conditions and how the follower interpreted and executed any previous instruction elements. This experiment measures the distribution of when the inferences are necessary. These are further defined in Section 5.5.2.

Pragmatic (discourse) cues have the least effect on performance. Without pragmatic cues, the performance impact is 16%. Overall, performance without pragmatic cues was ($M = 43.4\%$, $SEM = 1.2\%$), $N=1522$, different from the full performance at $t(1521) = 9.46$, $p \leq 0.001$. In Corpus 1, the impact was 17%; Corpus 2, 13%; and Corpus 3, 15%.

Without exploration procedures, the impact on MARCO’s overall performance is 21%. In Corpus 1, the impact was 23%; Corpus 2, 21%; and Corpus 3, 32%. Exploratory procedures include procedures to match the view description to the view when the view does not contain enough information to disambiguate. The canonical example of this is “Face the longer end of the hall.” Even if the follower is already facing the longer portion of the hallway, it cannot know its state without seeing or having seen what is behind it. Exploratory procedures also include recovery procedures when lost or when the described pose is visible in the distance.

Without recognizing the implicit procedures marked by syntactic cues, the impact is 38%. In Corpus 1, the impact was 41%; Corpus 2, 38%; and Corpus 3, 39%. This includes syntactic cues for both inferring a *Travel^p* implicit in a *Turn^p* command (“At the end of the hall, turn right.”) and a *Turn^p* in a *Travel^p* procedure (“With your back to the wall, walk forward.”).

Recognizing semantic cues had the largest impact, decreasing performance by almost 52%. In Corpus 1, the impact was 45%, Corpus 2, 61%; and Corpus 3, 62%. Some semantic cues are in the preconditions of the stated procedure, for instance, in “Take the green path,” a *Travel^p* is necessary when the green path is distant. Turns can also be implicit in a procedure’s preconditions, e.g. a *Turn^p* to face the path in “Go down the brick hallway.” An example of verb frame encoding a procedure is “Take the second left,” implying *Travel^p* forward to the second intersection that has a path exiting to the left.

See Appendix B.7 for more details of individual behaviors.

6.4.1 Implicit Procedure Cues Results by Rating

Ablating the various cues for implicit procedures had varying effects over the spectrum of instruction quality (Table 6.3). Exploration cues are particularly important for poor quality instructions (58% impact). Often the worst instructions had gaps and errors necessitating

Quality Range	1.0–2.5	2.5–3.5	3.5–5.0	5.0–6.0	All	Impt
People	35% [‡]	52% [‡]	73% [‡]	85% [‡]	64% [‡]	-20%
Marco	22%	33%	60%	79%*	52%	0%
– w/o Implicit Pragmatic	18% [†]	27%*	51% [†]	67%*	43% [‡]	16%
– w/o Implicit Exploration	9% [‡]	19% [‡]	49% [‡]	70%	40% [‡]	23%
– w/o Implicit Syntactic	13% [‡]	18% [‡]	37% [‡]	53% [‡]	32% [‡]	38%
– w/o Implicit Semantic	6% [‡]	11% [‡]	30% [‡]	46% [‡]	25% [‡]	52%

Table 6.3: Comparing performance of MARCO versions without different implicit procedure cues

exploration. However, exploration cues are rarely crucial to following the best-rated instructions, the impact is not significant impact at 4%.

Discourse pragmatics cues only had an impact of 8–19% across all quality instructions. However, the impact of pragmatic cues on the worst instructions was smaller (8%), significant at $t(268) = 2.53, p \leq 0.006$. Inferences from pragmatic cues on poor instructions may be more challenging or simply not yet implemented.

Semantic cues had a bigger impact on worse instructions than better instructions, although it had a significant impact across the quality spectrum of between 37–74%. The impact scaled up as the instructions worsened. Since semantic cues are domain or verb dependent, they may require more effort to interpret, and therefore maybe cause people to rate an instruction worse if it requires inferring implicit procedures from semantic cues.

Inferring implicit procedures from syntactic cues has a consistently high impact across the quality spectrum. On the best instructions, the impact is 26%; for each of good, mediocre, and poor instructions, the impact is between 38% and 46%. Since syntactic cues are domain-independent, they may not require as much effort to interpret as the domain-dependent semantic cues. This would explain why the impact of inferring implicit procedures from semantic cues is fairly evenly distributed across the quality spectrum.

Quality Range	1.0–2.5	2.5–3.5	3.5–5.0	5.0–6.0	All	Impt
People	35% [‡]	52% [‡]	73% [‡]	85% [‡]	64% [‡]	-20%
Marco	22%	33%	60%	79%*	52%	0%
Marco w/o Implicit Procedures	5% [‡]	7% [‡]	22% [‡]	40% [‡]	20% [‡]	62%

Table 6.4: Comparing the performance of MARCO without implicit turns and travels

6.5 Inferring Different Types of Implicit Procedures

Besides the different cues for implicit procedures, there are also different inference products — the implicit procedures themselves. This section looks at the impact of inferring only Turn^p procedures, only Travel^p procedures, inferring both types of procedures and inferring neither. This chapter is an update of (MacMahon et al., 2006) with a further developed MARCO model and a separate evaluation corpus.

The last section examined the impact of cues of implicit procedures and this section examines the impact of the results of inferring implicit procedures. We present results for five types of followers: (1) human participants, (2) the full MARCO model, (3) MARCO without Turn^p inference, (4) MARCO without Travel^p inference, and (5) MARCO without either Turn^p or Travel^p inference (only explicitly commanded procedures). For people, the results are the mean over runs from multiple participants following each instruction set, each beginning at the start location facing a random direction. For the MARCO cases, the presented results are the mean over four runs, facing each of the four directions at the start.

Without inferring Travel^p procedures, only implicit Turn^p procedures, MARCO’s performance drops to 32.5% ($SEM = 1.1\%$), an impact of 37%. Across the corpora, the impact is 29%, 52%, and 43%. In all corpora, the difference between the full MARCO and MARCO that did not infer Travel^p procedures is significant ($t(690) = 12.19, p \leq 0.001$; $t(431) = 11.76, p \leq 0.001$; $t(398) = 7.19, p \leq 0.001$).

If MARCO does not execute implicit Turn^p procedures, but only implicit Travel^p procedures, performance slips to 25.3% of the 1522 instructions followed by people,

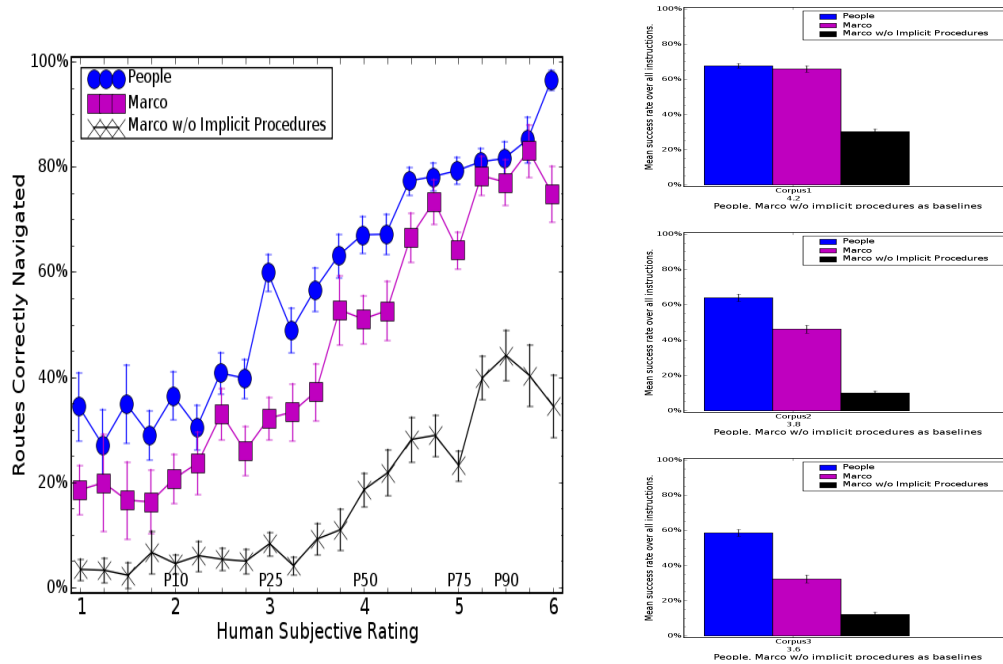


Figure 6.6: Implicit turns have a higher impact than implicit travels. Inferring turns literally gets the agent on the right path, where it gets new information.

$SEM = 0.9\%$. Across the corpora, the impact is 47%, 57%, and 54%, a very consistent drop-off in performance. In all corpora, the difference between the full MARCO and MARCO that did not infer $Turn^p$ procedures is significant ($t(690) = 20.02, p \leq 0.001$; $t(431) = 13.90, p \leq 0.001$; $t(398) = 9.56, p \leq 0.001$).

For all corpora, implicit $Travel^p$ procedures have less impact than implicit $Turn^p$ procedures. This difference was significant overall ($t(1521) = 7.34, p \leq 0.001$), and in Corpus 1 ($t(690) = 7.82, p \leq 0.001$) and Corpus 3 ($t(398) = 2.35, p \leq 0.010$), but not Corpus 2 ($t(431) = 1.25, p \leq 0.105$). It makes some sense for an implicit $Turn^p$ to be more crucial than an implicit $Travel^p$, as the $turn^a$ causal action changes the information available drastically, by bringing a new path into view. A $travel^a$ causal action, on the other hand does not change the information available to an agent with distal perception as much.

Following purely explicit instructions, without inferring either Turn^p or Travel^p procedures, MARCO can successfully follow just 19.8% of the routes in the 1522 instruction corpus. The effects of inferring implicit Turn^p or Travel^p procedures are neither fully independent nor fully dependent, both are critical for route instructions. Across the corpora, the impact varies from between 54% and 78%.

For all instructions, the difference is significant from both inferring only implicit Turn^p procedures ($t(1521) = 9.21, p \leq 0.001$) and Travel^p procedures ($t(1521) = 15.97, p \leq 0.001$). The difference from inferring only implicit Turn^p s is significant in Corpus 1 ($t(690) = 12.92, p \leq 0.001$); Corpus 2 ($t(431) = 8.00, p \leq 0.001$); and Corpus 3 (Turns $t(398) = 4.89, p \leq 0.001$). Inferring neither implicit Turn^p nor Travel^p procedures has a significant impact over inferring only implicit Travel^p procedures is significant in Corpus 1 ($t(690) = 5.23, p \leq 0.001$); Corpus 2 ($t(431) = 8.62, p \leq 0.001$); and Corpus 3 ($t(398) = 2.49, p \leq 0.007$).

6.5.1 Implicit Procedure Experiment Results by Rating

Inferring implicit procedures is essential for following the nearly all of the lowest rated instructions in this corpus, but still has an impact of nearly 50% even for following the highest rated instructions. Table 6.3 summarizes the results graphed in Figure 6.5 across broad classes of human *post hoc* subject instruction ratings. In this discussion, r will denote the mean rating on an instruction set from the six human followers.

For poor instructions, $r \leq 3.5$ out of 6, MARCO is effectively crippled without inferring implicit procedures, with an impact of 77% on instructions rated $1 \leq r \leq 2.5$ and 79% on instructions rated $2.5 < r \leq 3.5$. On good but not excellent instructions ($3.5 < r \leq 5.0$), MARCO can follow a significant number of instructions without inferring implicit procedures, but performs much better by inferring procedures (impact 79%), especially Turn^p procedures, (impact of 65%). Making an implicit turn^a can put the follower on the correct path, revealing a view down the path which shares very little information with views

facing in other directions. A *Travel^p* moves the agent to a new place, but does not bring as much new information into the view as the followers are able to see distant objects.

On the best instructions, those rated $r > 5$, the full MARCO system and people had only a small significant difference in performance, 7% $t(778) = 1.92, p \leq 0.028$. Without inferring implicit procedures, MARCO had significant decrease in performance even on the best instructions, with a 45% impact. Inferring implicit procedures accounts for nearly all the success on poor instructions and about half the success even on the best instructions.

All of these trends are consistent across all three corpora.

6.6 Object, Structural and Appearance Landmarks

Landmarks are crucial to any theory of route instructions. Landmarks are the reference points that directors use to describe the route and that followers use to orient themselves to the route. To navigate to the destination, the follower must match the landmark descriptions with observations seen along the way.

6.6.1 Review of Landmarks Types

Directors can refer to several types of landmarks. Stankiewicz and Kalia (2007) distinguish *object landmarks* and *structural landmarks*. *Object landmarks* are discrete objects in the environment that are independent of the navigational structure of the environment, such as the pictures on the walls and furniture on the floor. *Structural landmarks* are spatially extended and based on the navigational structure of the environment — how the paths connect and terminate — and include dead ends, T intersections, and descriptions of hallway length.

This distinction between object and structural landmarks is common. Hansen et al. (2006) split landmarks into Point, Linear and Area landmarks at their “Conceptual” level. Our object landmarks fall into their Point landmarks and structural landmarks are Linear or Area. Sandstrom et al. (1998) similarly split landmarks into “landmarks” (object

landmarks) and “geometric information” (structural landmarks). Sorrows and Hirtle (1999) distinguish visual, cognitive, and structural landmarks, which Raubal and Winter (2002) formalize into a computational model accounting for “visual, semantic, and structural” attributes. Their structural landmarks correspond to ours and our object landmarks fall into their broader visual landmark category. Siegel and White (1975) propose that these types of landmarks play primary roles at different stages in learning large-scale spatial environments.

In this analysis, we decompose landmarks by the complexity of the required perception. Like the hierarchical lattice of spatial ontologies in the Hybrid Spatial Semantic Hierarchy (HSSH) (Kuipers et al., 2004; Kuipers, 2000), simpler perceptual attributes and landmark types are required for more complex types. The landmark types examined here are

1. *Appearance Landmarks*: Characterized by a single perceptual attribute, including color (red, blue), luminance (dark, light), and texture (wooden, brick).
2. *Object Landmarks*: Discrete, localized objects, e.g. the furniture in some intersections (chair, sofa, lamp).
3. *Structural Landmarks*: Describe navigable space. We break these into
 - (a) *Causal Landmarks*: Local simple structural landmarks that form the components of more complex descriptions: walls, hallways and places.
 - (b) *Intersection Landmarks*: Compound structural landmarks describing the local topology, or shape, of intersections (T intersection, dead end, path to the right).

6.6.2 Landmark Recognition Ablation Study

To test the navigation impact of different types of landmarks in this route instruction corpus, we ran an ablation study removing the ability to recognize appearance, object, intersection,

Quality Range	1.0–2.5	2.5–3.5	3.5–5.0	5.0–6.0	All	Impt
People	35% [‡]	52% [‡]	73% [‡]	85% [‡]	64% [‡]	-20%
Marco	22%	33%	60%	79%*	52%	0%
– w/o Object Landmarks	15% [‡]	22% [‡]	46% [‡]	62% [‡]	39% [‡]	25%
– w/o Intersection Landmarks	15% [‡]	22% [‡]	46% [‡]	60% [‡]	38% [‡]	25%
– w/o Appearance Landmarks	13% [‡]	18% [‡]	39% [‡]	57% [‡]	33% [‡]	35%
– w/o Structural Landmarks	9% [‡]	11% [‡]	25% [‡]	37% [‡]	21% [‡]	58%

Table 6.5: Comparing the performance of MARCO without the ability to recognize different kinds of landmarks

and all structural landmarks. For this study, landmarks are any description of the world along the route — what MARCO models as view descriptions. In this study, we test how much the ability to recognize each type of landmark contributes to successfully following the route instructions.

Where we have ablated the landmark recognition, the recognition code always returns *True*, which allows matching the referring phrase by using the other types of landmark information. For instance, told to “Take the blue path,” without Structural landmarks (specifically, Causal landmarks), MARCO will face something blue (in this study, always a path). Given this description without Appearance landmarks, MARCO will face a path, whether it is blue or not. Without either Appearance or Structural landmarks, any view will match. Thus, a description incorporating at least two types of landmarks (e.g. “Face the chair on the rose hallway.”) may be robust to the loss of any one, depending on the environment and whether the additional information is elaborative (redundant) or contrastive (necessary).

In this study, the three types of landmarks were each ablated separately. Additionally, we tested the result of ablating only Intersection Structural landmarks, but not Causal landmarks. Recognizing Intersection landmarks without the Causal landmarks of paths and walls is not possible, by definition.

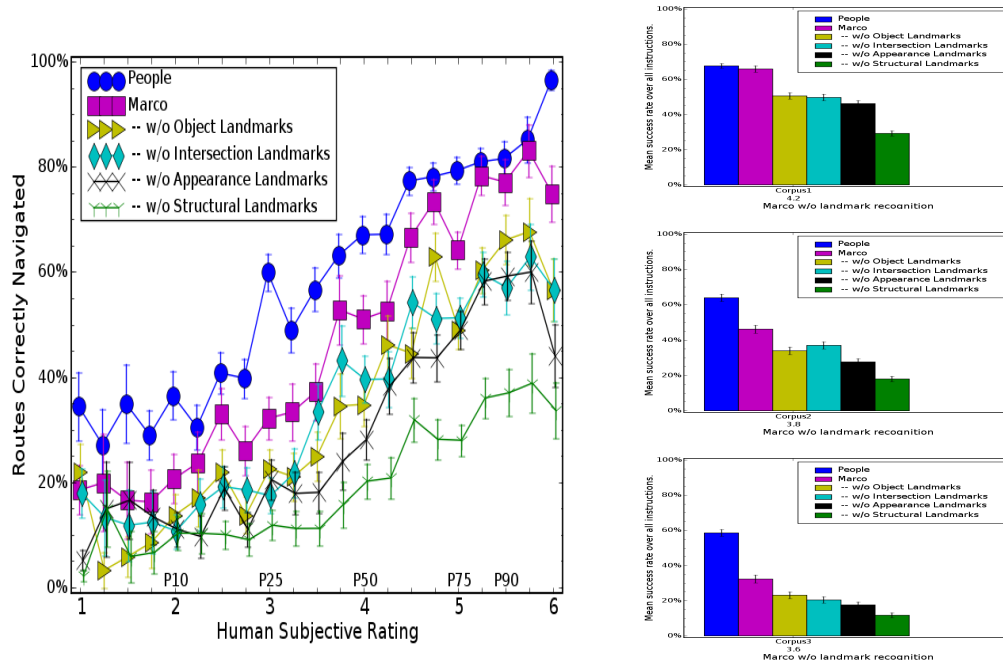


Figure 6.7: Success rates for MARCO without the ability to recognize types of landmarks.

6.6.3 Landmark Recognition Ablation Results

The impact of ablating the different landmark types accords with the predictions suggested by the HSSH and with previous psychological studies (Stankiewicz and Kalia, 2007). Structural landmarks, which encode information about the navigational structure of space, have the highest impact. Appearance and Object landmarks have the next highest impact, followed by Intersection landmarks alone.

Simpler landmarks and landmarks directly involved in the task have higher impact than more complex landmarks. This pattern of impact matches the expected pattern from the literature. This is evidence for a hierarchical ontology of spatial and perceptual attributes: the more fundamental entities of Causal landmarks have higher impact than landmarks built on them (Intersection) or based on attributes peripheral to the task (Object and Appearance). Structural landmarks refer to the navigational structure of space, and so the language of

route instructions is structured around Structural landmarks.

Over the entire tested corpus, the impact of running without the ability to Structural landmarks is 58%, i.e. the ablated model succeeds 42% as often as the full MARCO. On the best-rated instructions (mean rating 5.0–6.0), the impact is 49%. For middle-rated instructions, the impact is 59% and 67% for instructions rated 2.5–3.5 and 3.5–5.0, respectively. For the worst rated instructions, the impact is 60%.

Appearance landmarks, have less impact (35%), but more than any other type of landmark. Due to the prevalence of different floor textures and the textures being distributed along long straight paths, appearance references are extremely common in this study. Moreover, simple appearance references tend to be contrastive between the available paths. The impact is 22% for the best instructions, 34% for the good instructions, 47% for adequate instructions, and 44% impact for the worst instructions. Appearance landmarks are simple to perceive and describe.

Object landmarks have the next biggest impact in this study at 25%, although they are relatively sparsely distributed in the environments. Nearly every hallway has a distinct floor texture, giving a distinctive appearance. Every place can be described in terms of its intersection structure. However, only about 25% of the places in the tested environments have an object. Overall, the impact of object landmarks is more evenly distributed across the rating spectrum. For both the best instructions, the impact is about 14%, while for 3.5–5.0 rated instructions, the impact is 23% and for 2.5–3.5 rated instructions the impact is 34%, for the worst, the impact is 32%.

Intersection landmarks alone have about the same impact in this study as object landmarks. This can be explained by two factors. First, the local topology of the intersection (its shape) is a relatively complex concept. Second, many intersections are described by the appearance of their component paths in the instructions, and this is usually sufficient in these environments to uniquely identify an intersection. Intersection landmarks have an impact of 25% , with the two best-rated classes of instructions around

20% impact and the worst two classes of rated instructions, the impact around 33%. Note that many of the worst instructions simply state a description of the destination in terms like “Position 7 is at the intersection of the yellow and grey floored halls.” or “Position 4 is at the dead end of the yellow fish hall with yellow floors.”

All of these differences are consistent and significant across all three corpora. All are significant for all the rating groups of instructions and across all three corpora.

6.7 Hybrid Spatial Semantic Hierarchy

6.7.1 Review of the Hybrid Spatial Semantic Hierarchy

The original Spatial Semantic Hierarchy (SSH) (Kuipers, 2000, 2006; Remolina and Kuipers, 2004) represents large-scale space with four distinct representations: (1) *control laws* operating on perceptual attributes that allow reliable motion among *distinctive states*; (2) *causal schemas* that model movement between distinctive states and *views* that represent what can be observed at a state, which represent the local navigation affordances of the continuous world as a deterministic finite automaton; (3) a *topological* model consisting of *places*, *paths*, and *regions*, which represent the global navigation structure of the continuous world as a topological map or set of maps; and (4) *local metrical* information, such as the distances between places on a path and the angles between path segments at a place.

The Hybrid Spatial Semantic Hierarchy (HSSH) builds on the SSH by synthesizing a better representation of small-scale space (Beeson et al., 2007; Kuipers et al., 2004; Modayil et al., 2004). The *local perceptual map* (LPM) is a representation of the positions of objects, obstacles, and hazards in the space around the agent, such as an occupancy grid (Moravec and Elfes, 1985), built directly from ranged perception, such as a laser-range finder, sonar, or vision. The LPM represents the metrical layout of navigable space in the local surround of the agent. The *local topological map* describes a place by the

structure of its navigational affordances: how the agent can move through, in, and out of the place. Specifically, the local topology of a place can be represented by the *small-scale star* modeling the relationships among the path segments entering and exiting the place.

The Hybrid Spatial Semantic Hierarchy provides the representations needed to model the entities and actions needed for navigating guided by route instructions. The control laws at the local metrical provide the basic skills needed for locomotion. The causal procedures at the local topological level provide behaviors matching descriptions of turns (“Turn left” and “Take a right”) and travel procedures (“Go forward twice” and “Move to the next place”). The local topological level allows the description of intersections, e.g. “Take the second left” and “at the four way intersection of the red hall and the rose halls.” The global topological level is needed for reasoning about paths, such as that when told, “Follow the hallway around to the chair,” the follower should take forced-choice turns or default procedures to proceed along the hall until reaching the chair.

We make one distinction over previously published descriptions of the HSSH: the Causal level is split into open and closed-loop behaviors. In the control level of the SSH, open and closed-loop control laws (a.k.a trajectory-following and hill-climbing control laws) describe different behaviors on the same representations. Similarly, open-loop causal behaviors have a termination condition that is a relative offset, but do not have a description of the resulting pose. This distinction is in the route instruction literature, usually discussed as whether or not navigation and description strategies use landmarks (Allen, 2000; Brown et al., 1998; Burnett, 2000; Dabbs et al., 1988; Denis, 1997; Geldof, 2003; Jackson, 1998; Lawton, 2001; Lovelace et al., 1999; Michon and Denis, 2001; Raubal and Winter, 2002; Tenbrink et al., 2002; Tenbrink and Klippel, 2005; Ward et al., 1986).

6.7.2 Hybrid Spatial Semantic Hierarchy Ablation Experiment

We have run an experiment looking at the impact of the HSSH ontologies for following route instructions, building on the landmark recognition ablation study (6.6). Here, we have categorized the landmark recognition and navigation skills by the HSSH representations each requires, as detailed below.

Below, we introduce each category, and then name the behaviors and representations ablated for that HSSH level.

1. *Causal*: Causal control laws, plus causal representations and behaviors. *Causal Landmarks, Travel To Next, Travel Between Turns, Travel On Final Turn*
 - (a) *Open-loop causal control laws*: Simple turn^a (*Right\Left*) and travel *Forward* causal actions, *Distance Count*
 - (b) *Closed-loop causal control laws*: Repeat Causal action until view description is matched. *Travel Until, Face Description, Turn Until View, Object Landmarks*
2. *Local Metrical*: Building and reasoning with a model of the local sensory surround. *Travel To Distant View, View Memory, Perspective Taking, Face Distance*
3. *Local Topological*: Representations and reasoning about local navigational affordances. *Recognize Structural, Intersection Landmarks, Turn Toward Path, Face Distance, Face Until*
4. *Topological*: Representations and reasoning about places and paths. *Use Follow, Use Find Turn Between Travels, Travel Between Turns*

6.7.3 Hybrid Spatial Semantic Hierarchy Ablation Results

The route navigation in these environments is entirely dependent on the *Causal* level. Without *Causal* procedures and representations, hardly any of the instructions can be

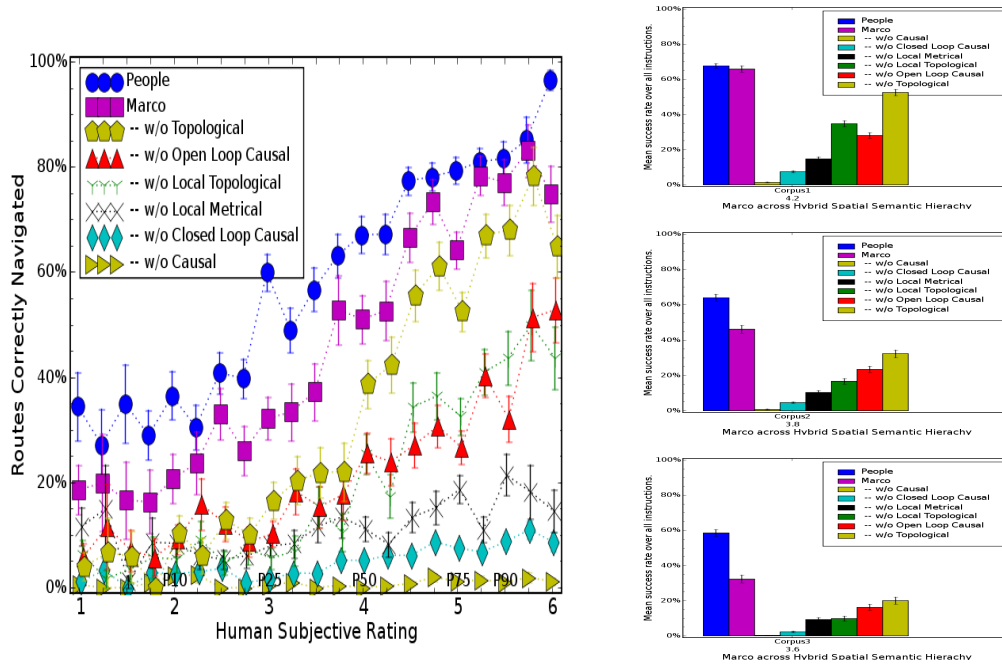


Figure 6.8: Route instructions depend on Hybrid Spatial Semantic Hierarchy representations with relative impact as the theory predicts.

followed, only about 1% of the corpus (98% impact). Causal representations model moving discretely between places on a path and paths at a place. They are fundamental to navigation though large-scale spaces.

Closed-Loop Causal behaviors have more impact (90%) than *Open-loop Causal* behaviors (54%). Part of this is that most of the instructions require a *Closed-Loop Causal* procedure to accomplish an initial orientation to the route, either with an explicitly commanded procedure, such as “Face the chair,” or an implicit one, e.g. “With your back to the wall, turn right.” *Closed-Loop Causal* procedures are used slightly more often over the entire corpus than *Open-loop Causal* procedures, on 702 of 771 instructions instead of 667 of 771 in Corpus 1 (including instructions that people did not follow). However, this is not enough to explain the discrepancy in impact. See Section 6.4.1

Quality Range	1.0–2.5	2.5–3.5	3.5–5.0	5.0–6.0	All	Impt
People	35% [‡]	52% [‡]	73% [‡]	85% [‡]	64% [‡]	-20%
Marco	22%	33%	60%	79%*	52%	0%
– w/o Topological	7% [‡]	18% [‡]	47% [‡]	69%	38% [‡]	26%
– w/o Open Loop Causal	9% [‡]	13% [‡]	25% [‡]	43% [‡]	24% [‡]	54%
– w/o Local Topological	6% [‡]	8% [‡]	27% [‡]	45% [‡]	23% [‡]	55%
– w/o Local Metrical	8% [‡]	8% [‡]	14% [‡]	16% [‡]	12% [‡]	77%
– w/o Closed Loop Causal	3% [‡]	2% [‡]	7% [‡]	9% [‡]	5% [‡]	90%
– w/o Causal	1% [‡]	0% [‡]	1% [‡]	1% [‡]	1% [‡]	98%

Table 6.6: Comparing the performance of MARCO variants without different levels of the Hybrid Spatial Semantic Hierarchy

for further discussion.

Ablation of the *Local Metrical* abilities has a high impact on route instructions following (77% impact). *Local Metrical* contains two crucial spatial reasoning skills: *View Memory* and *Perspective Taking*. *View Memory* controls the ability to remember views that are out of sight, as the agent turns in place and moves down a hallway. Without *View Memory*, the follower cannot form a local perceptual model of the local surround (81% impact) (Beeson et al., 2007; Kuipers et al., 2004; Modayil et al., 2004). Without *Perspective Taking*, an agent cannot use the local perceptual model to project what its view would be if it were at a different pose (67% impact) (Schober, 1993; Taylor and Tversky, 1996; Trafton et al., 2005).

Local Metrical spatial representations and reasoning skills allow the agent to move beyond purely reactive way-finding behaviors and behaviors with only impoverished state (e.g. maintaining a *Distance Count* over a *Travel^p*). *Local Metrical* behaviors have less impact on the worst instructions. Since these often describe only the destination, and not the route, reacting to immediate observations is usually sufficient, when these instructions are useful at all.

Local Topological has a significant but smaller impact, reliable across instruction quality (55% impact). The highest impact components of the *Local Topological* ablation

are *Intersection Landmarks* (26% impact) and *Face [Travel] Until* (33% impact) . In this option, the follower will turn to ensure that the destination of the current *Travel^p* procedure is visible to the front in the distance before starting to move forward. The follower uses *Perspective Taking* to project the view description of the *until* condition of a *Travel^p* into the distance, then executes a *Turn^p* procedure to face the projected view description. This is a *Local Topological* behavior, because it relies on the topological knowledge that the destination should be in front of the traveler and either visible or at least, possible, if the traveler can only see a limited distance along the path.

Without Local Topological Models (Beeson et al., 2007; Kuipers et al., 2004; Modayil et al., 2004), the agent cannot categorize intersections into common types, such as dead ends, four-way intersections, and corners. This has the highest impact within the instruction corpus for the worst instructions, where the director may rely on intersection shape alone to distinguish it from other places along a hallway.

Finally, the *Topological* ontology has a significant impact of 26%. The highest impact components of the *Topological* level are the high-level procedures *Find^p* and *Follow^p* and the *Travel to Next* heuristic. *Find* is the procedure that does undirected search until the desired view is visible in the distance, then executes a *Travel^p* procedure to reach it (impact 17%). *Follow^p* makes forced-choice turns along a constrained path (impact 4%). *Travel to Next* goes to the next match when a *Travel Until* is matched directly after an explicit *Turn^p* (impact 9%), e.g. “Turn right. Go to the corner,” when the right turn is also at a corner.

Overall, the ontologies line up in impact as the HSSH theory predicts, with the more lower levels having larger impact than the higher levels. The *Causal* level is fundamental to travel through large-scale spaces. Causal behaviors can be described by both open and closed-loop descriptions, but closed-loop behaviors are more reliable for both director and follower. *Local Metrical* representations allow the follower to model the local layout and projecting what the view would be after moving to another place within

the view, perspective taking. This allows the director to describe the world with a richer vocabulary, which reduces the amount of poses that are *linguistically aliased* – that can only be described in the same ways. The *Local Topological* level brings a common and concise vocabulary of intersection types, which allows the description of places as a whole, instead of with longer constructions detailing the positions of the component walls and paths. Finally, the *Topological* level allows the director to rely on the follower to infer unstated turns, reducing the communication cost without greatly reducing the likelihood of reaching the destination.

6.7.4 Extensions to the Hybrid Spatial Semantic Hierarchy

The work extending the Spatial Semantic Hierarchy to the HSSH takes advantage of improved sensors, processors, and algorithms to add models of small-scale space to the SSH, which only modeled large-scale space. The ability to reliably map and localize in the immediate sensory surround greatly reduces problems of mapping globally (Kuipers et al., 2004; Modayil et al., 2004). In the SSH, observations were abstracted to *views*, an opaque representation of what the agent saw at a pose. In the HSSH, observations at a place are combined into a rich model of the metrical shape and topological navigational affordances (how the agent can exit a place).

This dissertation suggests that the Hybrid Spatial Semantic Hierarchy will need to be similarly extended, to include models of *medium-scale space*. The ontology and global maps of large-scale space covers any space larger than the sensory horizon. The ontology and local maps of small-scale space cover the local place neighborhood immediately around the agent, that the agent can fully perceive with turns or local motion, while remaining completely localized.

The experiments in this dissertation show that people often rely on perception and reasoning about medium-scale space – distant places, visible and accessible from the immediate place. The route instruction director relies on the follower to be able to

perceive and reason about distant places that are partially visible. Medium-scale spatial reasoning and perception are required to handle explicit instructions, such as “Go toward the end of the hall with the chair,” and implicit procedures for ensuring that the destination is either visible on the path ahead, or that the path continues out of view.

6.8 Grammar Cross-validation

6.8.1 Review of Cross-Validation Methodology

Cross-validation is an evaluation technique to test machine learning systems (Mitchell, 1997). In cross-validation a large corpus of labeled examples is repeatedly partitioned into discrete training and test sets. For each pair of sets, the learning algorithm is given both the data and labels in the training set to learn the model and then evaluated by labeling the test set. This process is repeated for different partitions of the data, so that each example in the corpus is used several times in both different training and test sets. This process ensures that the split of examples into training and test sets does not bias the results, as well as measuring the variability across runs.

One commonly used cross-validation methodology is “leave-one-out” cross validation. The data set is shuffled and split into equal parts, by some criteria. For instance, we will perform cross-validation by splitting the corpus by the instructions’ directors and by the map the route is in. In “leave-one-out” cross-validation, the training set is all of the data other than one group (e.g. one director or map). The left-out group is the test set. The process is then repeated, using each group as the test set, with the remainder comprising the training set. After training and testing on all partitions of the data, the results are averaged across runs to give the mean performance and variability.

6.8.2 MARCO Grammar Cross-Validation Experiments

The methodology of cross-validation allows us to test the generalization of the grammar and parser under several different scenarios. We can test how consistent the grammar is across directors and across environments. Rather than splitting the route instruction corpus into training and test partitions randomly, we can split the corpus in controlled manners. We have run experiments with the corpus split by environment and by director.

In each of these studies, the Probabilistic Context-Free Grammar is trained on part of the corpus and then the MARCO agent is run using the output of the parser. All other parts of the agent remain the same, only the parse trees are generated by the trained grammar, instead of loaded from hand-corrected trees. Both of these studies were run only on Corpus 1.

For the cross-validation over maps study, the agent is trained on instructions from all directors from two of the environments and then run on the instructions from the other environment. This tests how much variability there is in the grammar across environments. Additionally, it tests the robustness of the agent to imperfect parses.

In the cross-validation over directors study, the grammar is trained on instructions from five of the directors and then tested on the instructions. This tests how variable the instruction grammar is across the different directors. This is one measure of how similar the styles of the directors are, at least at the surface level. This gives some prediction of how well the system will do on instructions from a new director.

6.8.3 MARCO Grammar Cross-Validation Results

Overall, the cross-validation experiments show the system is fairly robust across environments and across all but one of the Corpus 1 directors. With cross-validation over maps, MARCO succeeds on 51% on the corpus, a performance impact of 24% over using the gold-standard parses. With cross-validation over directors, MARCO succeeds on 38% on the corpus, a performance impact of 43% over using the gold-standard parses. Table 6.7

Quality Range	1.0–2.5	2.5–3.5	3.5–5.0	5.0–6.0	All
Human	33%*	46%*	75%	85% [†]	69%
Marco	25%	38%	73%	89%	67%
– w/ Cross-validation over Maps	22%	26% [‡]	55% [‡]	70% [‡]	51% [‡]
– w/ Cross-validation over Directors	17%*	25% [‡]	41% [‡]	48% [‡]	38% [‡]

Table 6.7: Performance on cross-validation runs by subjective rating.

and Figure 6.9 show the results across the instruction quality spectrum.

Note that in the directors cross-validation study, the PCFG actually trains on five-sixths of the corpus, while in the map cross-validation study, each training set is two-thirds of the corpus. Despite have more training examples in the director cross-validation study, the system performs significantly better in the map cross-validation. This is due to smaller differences in language use for directors between environments than between directors.

In the map cross-validation study, the highest rated two classes of instructions each have about the same performance impact from applying parsing using a learned, imperfect syntactic model as using the hand-corrected gold-standard parses, at about 24%. For instructions rated 2.5–3.5, parsing from the model learned on the other environments has a larger impact (34%), perhaps because when the director is unsure of the route, the syntactic style changes and becomes more variable to account for differing kinds of uncertainty. On the other hand, for the worst instructions (1.0–2.5), there is no significant impact from applying the learned model. When a director can only give minimal guidance, usually he or she is only describing the destination, which may be done in the same ways across environments.

In the director cross-validation study, the impact is greater across differing quality levels, but more consistent. Again the impact on the two best rated classes is similar, at about 45%. Here, the impact on the worse rated instructions is higher, at 34% for the 2.5–3.5 rated instructions and 28% for the lowest rated class, all significant differences from the gold-standard parse.

Figure 6.10 shows the performance of the cross-validation learned models for each

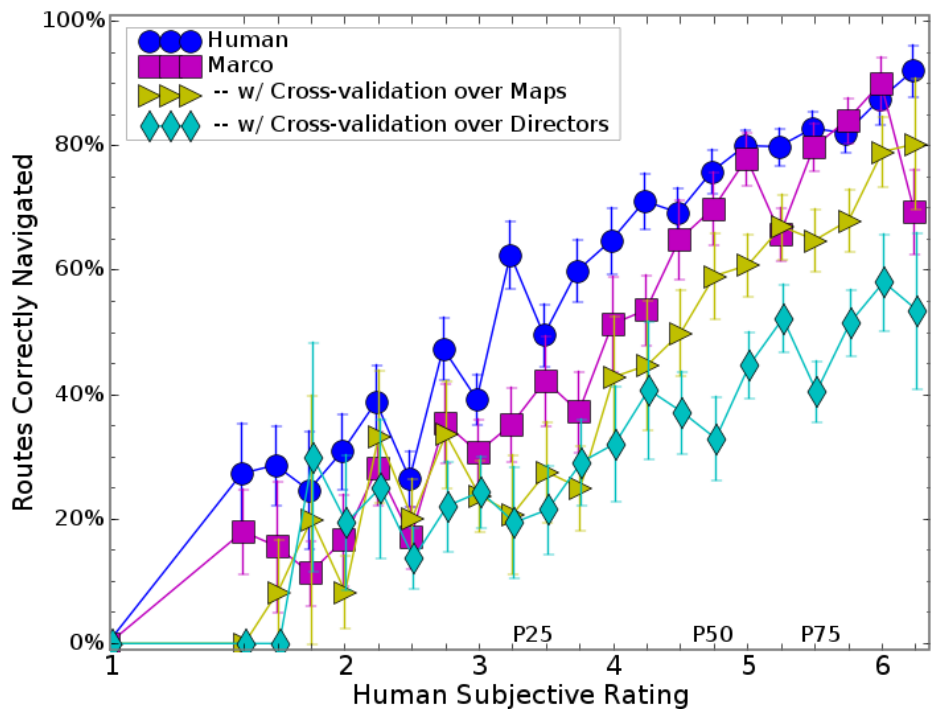


Figure 6.9: Success rates for MARCO under cross-validation across instruction quality.

director. The performance varies considerably across directors, as some directors have a distinctive style, using sentence structures not seen in the other directors. For instance, EDA tends to use a simple but unique sentence structure of “walk forward COUNT.” For example, no other director in Corpus 1 uses the word “twice” and EDA uses it in over half (67) of his instructions. EDA is very consistent across environments, with the map cross-validation parser yielding a 72% performance on his instructions, only an 11% impact off of the gold-standard parses. However, in the director cross-validation, MARCO has a 61% impact on EDA’s instructions from the gold-standard, and a 61% impact even from the map cross-validation parser. On all instructions from directors other than EDA, the impact of learning the syntax from other directors is only 37% compared to the gold-standard parses and 13% off of the performance of MARCO with the parser trained on instructions from all

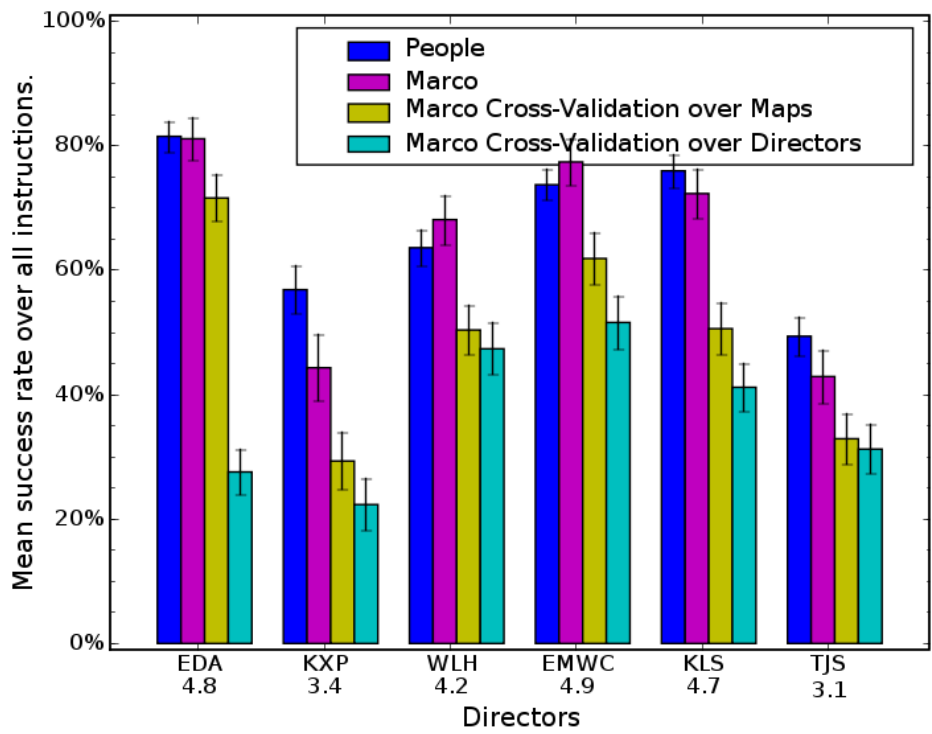


Figure 6.10: Success rates for MARCO under cross-validation per director.

directors in the other two environments.

Two of the directors, TJS and WLH, share almost all of their important syntactic variation with other directors. These directors only had a performance impact of 5% and 6%, respectively, comparing the MARCO’s performance with a grammar trained on instructions from the other directors against with a grammar trained on instructions from the other environments. The other three directors had intermediate impacts (17-24%), measuring a drop in performance from the map cross-validation to the director cross-validation. Only EDA (61%) was above the mean impact (26%) for this comparison.

6.8.4 Cross-Validation Discussion

These results, and the results of running on Corpus 2 and Corpus 3, show how MARCO handles unseen route instructions, using a naïve, PCFG parser (see Section 5.2). The other experiments in this chapter show how MARCO performs on the gold-standard hand-corrected parse trees. These two extremes — the naïve parser and the gold-standard parser — provide bounds for the expected performance of MARCO with a state-of-the-art parser. With a better parser or more training, MARCO should perform closer to the gold-standard performance level and certainly better than our naïve parser. Section 5.2 discusses more complex parsers that might be incorporated.

The second conclusion to draw from the cross-validation studies is that the idiosyncratic individual differences between directors have a higher impact than the differences from the same set of directors between environments. Training for a particular user would allow the system to learn his or her quirks in sentence structure, word usage, and implicit procedures. On the other hand, the cross-director performance impact should be mitigated as the system has trained with more types of users.

MARCO performs adequately even with a naïve parser and partial training on some of the instructions. The performance profile across instruction quality has the same profile as the performance on gold-standard parses, of performance scaling linearly with quality, albeit at a lower level. Part of the parsing problem, especially across individuals, is that the grammar training program makes no effort to use techniques to generalize the grammar from the instances in the training set, handling optional arguments and arguments with arbitrary order (Goan et al., 1996; Li and Abe, 1998; McClosky et al., 2006).

Even the best parser will not have 100% accuracy; the state-of-the-art is just higher than 90% per sentence (McClosky et al., 2006). When interpreting a paragraph of route instructions, there is a chance that each utterance will be mis-interpreted, and that chance accumulates over the instruction set. Even one mis-parsed sentence can throw off the execution, even with something as simple as an attachment error. To use a classic example,

if told to “Look for the man with a telescope,” the follower will fail if looking for “a man with a telescope” instead of through a telescope or vice versa. This points out the need for a parser that can re-rank the possible parses in response to linguistic, spatial, and task context (Chang and Mok, 2006) (See also Section 5.2). An orthogonal approach uses dialogue acts to clarify or correct the parses, which complements the methods in MARCO and better parsers (MacMahon, 2005; Weng et al., 2006).

6.9 Human-MARCO Discrepancy Analysis

Finally we want to compare the performance of people and the full MARCO agent on particular instructions. We closely examined the outcome of a random selection of route instructions from all three corpora. For each director, three instructions were selected at random from the set of all instructions which were successfully followed to the intended destination by either on least one trial by a human follower or MARCO. This section presents the results of comparing the performance of the full MARCO agent and people following these instructions.

Overall, since there were 30 followers in the three corpora, there were 90 instructions in the sample. Of these 90 instructions, MARCO and people performed about the same, within $\pm 25\%$ success rate, on 46 of the 90 instructions. On 31, people outperformed MARCO, but on 13 instructions MARCO reached the destination with a success rate more than 25 percentage points higher than people.

When comparing the success rates, remember we are comparing the mean success rate of a *population* of human users with many strategies and individual differences against one version of MARCO running repeatedly. If tested repeatedly, individual people might show the same pattern as MARCO of following some instructions reliably and failing reliably on others. Likewise, other variants of MARCO perform differently, and occasionally better on particular instructions, especially MARCO variants with different heuristics or parameters for accepting loose matches of descriptions.

<i>Rtg</i>	<i>Ppl</i>	MAR	<i>Direct</i>	<i>Map</i>	<i>S</i>	<i>T</i>	<i>Note</i>
5.1	100	100	EDA	Compact0	6	3	Same route
4.9	88	100	EDA	Sparse0	1	3	Same route
5.4	88	100	EDA	Sparse0	4	7	Same route
4.7	57	0	EMW	Sparse0	5	3	Missed locative phrase
							empty intersection between two which contain objects
4.6	86	100	EMW	Medium0	3	6	Same route
5.2	80	100	EMW	Medium0	7	5	Same route
5.5	100	100	KLS	Compact0	4	1	Same route
5.6	100	100	KLS	Compact0	4	5	Same route
5	100	100	KLS	Sparse0	7	1	Same route
4.7	86	100	KXP	Sparse0	6	7	Same route, one overshoot
2.5	25	50	KXP	Medium0	4	1	No orient, few landmarks
4.4	89	0	KXP	Medium0	4	7	Missed intersection phrase
							stop when the first fish hallway to your left appears
4.4	100	100	TJS	Compact0	1	5	Unnecessary search
2.7	86	75	TJS	Sparse0	3	4	No route description
4.9	100	100	TJS	Medium0	2	7	Same route
5.4	100	100	WLH	Sparse0	2	4	Same route
2.6	63	100	WLH	Sparse0	4	2	Recovers from missed anaphora
3.9	78	100	WLH	Medium0	2	6	Same route
4.4	82	82	Mean	All			

Table 6.8: Discrepancy analysis for sampled instructions from Corpus 1. The *Rtg* column shows the post-hoc human rating for the instruction text. The *Ppl* and MAR show the mean success rate for people and MARCO, respectively, following the instruction text. The *Direct* column shows the identifier for the director, *Map* is which environment the route traversed, *S* is the starting location and *T* is the target location.

In Corpus 1, the development corpus, there were only two instructions where people succeeded much more often than MARCO (See Table 6.8). In both cases, MARCO was not able to interpret a rare, complex phrase describing a location. In the first case, the empty intersection between two which contain objects, MARCO misses that the count two functions refers to other intersections, which each contain an object. In the second, MARCO interprets appears too literally, as when the path is visible in the distance, not waiting until it is immediately to the left. In each case, the view description representation could represent the same concepts, and in fact, the instruction modeler

accurately modeled other phrasings of the similar descriptions. On one route instruction set in this sample, MARCO out-performed people by more than 25 percentage points. In this case, there is a confusing part about halfway through the instructions, where some of the human followers give up.

In Corpus 2, after running MARCO as developed using Corpus 1, we added heuristics, language recognition code, and an attribute to the view description representation to follow more instructions. As Table 6.9 shows four instructions in this sample that the amended MARCO follows with a success rate more than 25 percentage points greater than people's success rate. Without the post-Corpus 2 modifications to MARCO, MARCO follows 7 of the instructions less often, and 4 more often, for a total of a 9% success rate gain on this sample (8% impact over all instructions in all corpora).

People succeeded on 13 of the sampled instructions much more often than MARCO and 19 instructions that were followed at about the same success rate. However, since MARCO fails consistently on some instructions that people follow extremely reliably, MARCO's mean success rate on this sample is 26 percentage points below people's. Note that the instructions MARCO failed on more often are mostly poorly rated, with a mean rating of 3.7, vs 4.2 for instructions with approximately the same success rates, and 3.3 for instructions where MARCO succeeds much more often.

Where people did better on Corpus 2 instructions, in six cases, MARCO did not correctly model a rarely occurring phrase, split evenly between phrases describing intersections, distances, and other spatial relations. In four cases, MARCO did not properly model the discourse by combining information across utterances. Most of these combined initial descriptions with elaborations. This sort of discourse reasoning is complex, but necessary to achieve human performance levels. The remaining three cases are from an un-modeled word, a mis-parsed sentence, and from instructions where people were able to recover from a lack of initial orientation despite few landmarks in the route description.

Table 6.10 shows the discrepancy analysis for Corpus 3. For this corpus, MARCO

was run on hand-corrected parses, with the Corpus 2 enhancements, but no modifications after examining Corpus 3. Of the 36 instructions in the sample, MARCO and people succeed within 25 percentage points on 12, exactly one-third. People succeeded more often on 16 of the sampled instructions and MARCO succeeded more often on the remaining 8. Here, there is not a significant difference between the ratings among these groups.

Of the 16 instructions where people out-perform MARCO, six are from discourse modeling errors or gaps, four are from phrase modeling errors or gaps, three are from missing heuristics for under-specified routes, two are from missed words, and one each are from a missed sentence and a missed anaphorical reference. For the discourse modeling errors, four are missing elaboration of descriptions by separate utterances and two are interpreting a redundant ‘stop’ command as an implicit *Travel!*.

Of the phrase modeling errors, two are complex descriptions of intersections and the other is a complex locative expression. In three cases, people are able to follow an under-specified route more often than MARCO, although in another three, MARCO is more successful. In three cases (including one of the elaboration cases), MARCO interprets a command more literally than people, by rejecting a L intersection as not matching the description “dead-end” and by not matching a coat-rack near but not at the end of the hall as “the end where there is a coat hanger.” Finally, in one case, MARCO recognizes the reference “the other” as referring to an easel, but does not recognize that it must find a different easel than the one in view.

6.10 Comparison to related work

MARCO is an embedded agent experiencing the world from a first person perspective, and therefore sometimes needs to act to gain information about unseen parts of the world – not just localize itself in a completely known world model. This differentiates the problem from the “kidnapped robot problem” of localization to a known map in large-scale space (Kaelbling et al., 1998) and from relating spatial language to a map (Levit and Roy, 2007) or

other scene within the perceptual horizon (Roy, 2005; Skubic et al., 2004b; Yu and Ballard, 2004). However, to build a full system to follow arbitrary instructions, MARCO would need to be extended to incorporate modules to represent, communicate, and reason about such small-scale spatial relations and actions.

Tellex and Roy implement spatial routines that achieve the preconditions of commands within the perceptual surround of a robot Tellex and Roy (2006, 2007), for instance taking the spatial context into account to move to the next opening before executing a “Go right” command. However, this work only accounts for single commands and does not test when the implicit actions are necessary in the linguistic context of a stream of instructions. We share the same general approach as Tellex and Roy – grounding a semantic parse of spatial instructions into a hand-coded procedure that executes autonomously. We have focused on the conditionals, complex commands, and series of commands across large-scale space. Tellex and Roy have focused on executing single commands across local small-scale space and spatial referring expression. Small-scale spatial relations are simply implemented in MARCO, while spatial routines are a more principled model of small-scale spatial relations, especially recognition.

6.10.1 Comparison to the Instruction-Based Learning project

MARCO has several significant differences from Bugmann and colleague’s Instruction-Based Learning (IBL) project (See Section 2.4.4, Bugmann et al. (2004)). These ablation experiments allow us to measure how often these differences affect successfully following the route instructions. One difference is that the IBL agent did not combine information across utterances. Overall, the pragmatic cues for implicit procedures that MARCO recognizes, but the IBL agent does not, account for 15% of the successful runs of MARCO on this corpus. The IBL agent also lacks the ability to reason about some of the local topology and local metrical entities, according to the examples the authors list outside the IBL agent’s capabilities (Bugmann et al., 2004). For MARCO, these capabilities together

account for a 4% impact on these corpora. This is a conservative estimate of the differences between MARCO and the IBL agent.

Bugmann et al. (2004) compared human and software instruction following performance. This work compared the performance for a robot navigating through a tabletop model environment given (1) an automated system translating the instructions from speech into programs, (2) an automated system following programs translated by hand from speech, and (3) people following the same instructions. People were able to reach the destination on 83% of the instructions, the robot followed hand-translated programs on 63% of the routes, and 14% of the routes automatically translated into programs “would actually lead the robot to the goal.”

Though our success rates are not directly comparable, since they start with raw speech and control a physical robot, our automated success rates are much more similar to our human rates. Their environment had fewer places, paths, and strong visual features than ours, but had more diverse intersections in a realistic town street layout. Their basic instruction-following method is similar to our work, but seems less robust to errors and omissions in the instructions, due to the spatial and linguistic knowledge we model.

The work in this paper is more easily and less expensively replicated, since no special robotic equipment or physical town model is needed. More importantly, our subjects learned the environments from the same first-person perspective as the human and software agents following the instructions and wrote instructions from memory. Bugmann’s participants only saw an outside, panoramic perspective of the town model while directing. This difference in how environments are learned and perceived between the directors and followers leads to a class of errors not present in our approach. Specifically, directors may refer to information unavailable to followers. Conversely, while our directors may make errors while learning the map through navigation or recalling the map while directing, these errors are cognitively interesting and prevalent in the real world. (See further discussion in Chapter 3.)

Like the IBL work, we find route instructions require an open set of procedures to execute. For deployed systems, this implies several options for robust interaction. The first option is to train its users to use a restricted set of actions, relations, and entities with a limited vocabulary and grammar. This is viable for long-term users of systems, especially professionals, such as urban search and rescue first responders and astronauts, and motivated users such as the disabled. For a robot or computer service that must interact with untrained users, one option is for the system to use dialogue actions to clarify or rephrase utterances that are not understood (MacMahon, 2005).

MARCO does not yet model certain types of linguistic behaviors that are important to instructions, such as verb aspect (Narayanan, 1997) or negative imperatives (Vander Linden and Di Eugenio, 1996). Finally, systems could use learning techniques and context information to learn the meanings of unknown words (Regier and Carlson, 2001; Roy, 2005; Siskind, 1990, 1995; Yu and Ballard, 2004), and perhaps programming-by-demonstration to learn new procedures (Jung et al., 2006; Nicolescu and Mataric, 2003).

6.11 Conclusions from MARCO experiments

Overall, one essential finding of this dissertation is the impact of implicit procedures in following complex procedural instructions. Inferring implicit procedures is just as important to the success of following spatial route instructions as recognizing and executing each type of explicitly commanded basic open- and closed-loop turn and travel procedures. Further, this dissertation proposes four kinds of cues to implicit procedures that will occur in all kinds of instructions: syntactic cues, semantic cues, pragmatic cues, and exploratory procedures. We measure how often each of these are necessary to follow spatial route instructions, finding that syntactic and semantic cues have the highest impact, but pragmatic and exploratory cues also are frequently crucial to infer the correct procedure from under-specified instructions.

Our ablation experiment of different spatial skills and representations at the

different levels of the Hybrid Spatial Semantic Hierarchy provides empirical support for the theory. The experiments first show that all of the variety of spatial ontologies are necessary for following spatial route instructions. Second, the HSSH theory predicts a hierarchy of representations built up from lower levels. Our experiments find the impact of ablating representation and reasoning at each level matches the predicted order of the spatial models, providing empirical evidence that the model captures essential characteristics of human spatial cognition.

The discrepancy analysis experiment shows several areas for improvement in MARCO. First, MARCO needs to better combine information between utterances, especially by modeling which utterances are elaborations of others. For instance, people, but not MARCO, combine this information in “Position 3 is located in the hallway of butterflies. It has wood floors all around it.” Second, MARCO should handle more spatial references, perhaps by building on top of work such as (Blisard et al., 2006; Klippel and Winter, 2005; Perzanowski et al., 2003; Skubic et al., 2001, 2004b; Tellex and Roy, 2006). Third, MARCO should have a mechanism for handling small errors in landmark descriptions, such as mis-characterizing an intersection, mis-remembering an object, or approximating a spatial relation. Adding in a best-fit match of the view description could also handle stochastic perception on a mobile robot, with noisy sensors.

5	0	25	BKW	Medium1	3	5	Wrong turn direction
5.2	100	100	BKW	Medium1	7	2	Same route
5.3	100	100	BKW	Medium1	7	6	Same route
4	100	100	BLO	Compact0	5	6	Same route
3.5	100	0	BLO	Compact0	6	3	Imprecise: Dead end is T intersection
3	100	0	BLO	Compact0	7	6	Missed attachment of distance
							Go straight down the red brick floor two stops...
5.3	100	0	JJL	Compact0	3	1	Missed implicit turn between travels
3.5	100	100	JJL	Compact0	3	7	Same route
3	100	100	JJL	Compact0	6	2	Same route
2.2	17	50	JNN	Sparse0	3	2	Confusing route description
5	100	100	JNN	Sparse0	3	4	Same route
3.7	67	100	JNN	Sparse0	7	3	Same route, some overshoot
4.3	67	75	JXF	Compact1	2	5	Handles negation
4.8	100	75	JXF	Compact1	7	3	Extra search w/o phrase
3.2	75	100	JXF	Compact1	7	6	Same route, unknown word 'crack'
3	100	0	LEN	Compact1	2	4	Not modeling discourse elaboration
5	100	25	LEN	Compact1	6	1	Not modeling discourse elaboration
3.5	100	0	LEN	Compact1	7	6	Missed distance phrase
							The last intersection before the T intersection
3.8	83	100	MJB	Sparse1	3	4	Same route
6	100	0	MJB	Sparse1	3	7	No model for locative phrase
							on the square just past the first segment of grass.
5.2	100	100	MJB	Sparse1	4	1	Same route
5	100	100	MXM	Sparse0	3	4	Same route
1.8	40	75	MXM	Sparse0	4	7	Underspecified, ambiguous route
1.9	14	0	MXM	Sparse0	7	5	Underspecified, ambiguous route
4	100	100	MXP	Medium0	4	3	Same route
3	100	0	MXP	Medium0	4	5	No model of locative phrase
							the first set of butterfly pictures, 2 facing each other
2.3	67	25	MXP	Medium0	7	4	No orientation, no landmarks
5.5	50	0	PXL	Medium0	2	3	No model of locative phrase
							until where you hit a gray wall in one step
2	100	0	PXL	Medium0	2	5	Mis-model distance phrase
							one step away from the direction of the painting stand.
2.5	100	25	PXL	Medium0	3	4	Mis-model intersection phrase
							at a 3 street corssing in which one floor is designed ...
5	100	100	QNL	Medium1	4	6	Same route
5.5	50	100	QNL	Medium1	5	4	Reference to turn dir, distance error
5.7	100	100	QNL	Medium1	6	7	Recovers from misinterpreting phrase
1.8	50	25	TXG	Sparse1	3	1	Underspecified route, Position landmarks
4	100	50	TXG	Sparse1	4	2	Named landmark as reference
3.3	100	100	TXG	Sparse1	6	2	Same route

Table 6.9: Discrepancy analysis for sampled instructions from Corpus 2.

3.5	75	0	ARL	Sparse1	3	6	No route, imprecise: dead end is an L int.
3	100	0	ARL	Sparse1	4	3	No route, discourse elaboration missed
3	100	0	ARL	Sparse1	5	6	No route, imprecise: dead end is an L int.
1	100	0	JLM	Compact0	1	2	Missed anaphora
							You are at one easel. Find the other.
4	100	0	JLM	Compact0	1	7	Discourse elaboration not modeled
							the four way intersection that meets these specifications.
4	100	100	JLM	Compact0	4	2	Extra turn, travel ignored after termination
4	0	25	JTM	Sparse0	3	4	Distance error
3	50	0	JTM	Sparse0	4	1	Extra travel for redundant stop
							move forward twice and stop
5	50	0	JTM	Sparse0	6	5	Extra travel for Move once and stop.
2	50	75	JXL	Compact1	2	4	Missing turn in instructions
2	100	100	JXL	Compact1	4	5	Negative advice
3	50	0	JXL	Compact1	7	2	Ambiguous sentence
							until the end where there is a coat hanger. (not quite at end)
4	50	100	KAJ	Compact1	4	2	Same route, some undershot
5.7	67	100	KAJ	Compact1	4	6	Same route, some undershot
2	33	100	KAJ	Compact1	7	5	Recovers from wrong turn direction
4.5	50	100	KXK	Medium0	3	7	Same route, some make extra turn
3	75	100	KXK	Medium0	7	4	Same route
2.5	25	0	KXK	Medium0	7	6	Underspecified route
1	0	50	LCT	Medium0	1	7	Underspecified route
4	50	0	LCT	Medium0	4	5	Underspecified route
1	100	0	LCT	Medium0	6	4	Missed discourse elaboration
4	50	100	MHH	Sparse1	4	1	Same route, some undershoot
3.5	50	0	MHH	Sparse1	4	5	Missed locative phrase
							until the left right before the yellow
1	50	50	MHH	Sparse1	7	4	Missing turns in instructions
2.7	33	0	RRE	Sparse0	1	5	Incongruent, ambiguous route
1	0	75	RRE	Sparse0	1	7	Incongruent, ambiguous route
2.5	50	0	RRE	Sparse0	5	4	Incongruent, ambiguous route
2.5	50	0	SCD	Sparse0	2	6	Missed intersection phrase
							Walk to the back of this corridor.
5.5	100	100	SCD	Sparse0	5	3	Same Route, reference between them
3	100	75	SCD	Sparse0	7	3	Missed phrase
3	100	100	SMA	Medium1	4	5	No route, only destination description
5	100	0	SMA	Medium1	5	1	Missed intersection phrase
							paths described as left and right are perpendicular
2.5	33	100	SMA	Medium1	7	6	No route, only destination description
4.7	100	100	WAB	Compact0	6	2	Same route
4.7	100	100	WAB	Compact0	6	4	Same route
5	100	0	WAB	Compact0	6	5	Discourse elaboration, Missed phrase
							last set of blue tiles

Table 6.10: Discrepancy analysis for sampled instructions from Corpus 3.

Chapter 7

Conclusions

This dissertation presents the language and task corpus methodology, which investigates what is necessary to understand natural instructions in order to follow them as well as people do. We applied the language and task corpus methodology to spatial route instructions, tying together a novel instruction corpus, navigable environments, and human and artificial agents performing complex linguistic and spatial reasoning tasks. We present a software system, MARCO, that approaches human levels of performance in applying instruction texts to navigate a described route through an unknown, large-scale space. Comparing the performance of MARCO model variants, we find inferring implicit procedures, in addition to executing explicitly commanded procedures, is essential to following poorly-rated instructions and crucial even on about half of highly-rated instructions.

This dissertation shows the Hybrid Spatial Semantic Hierarchy (HSSH) applied to both human-robot interaction and to perform complex navigation tasks (Sections 6.7 and 5.8). We show the HSSH hierarchy of spatial representations are both sufficient and necessary to represent the procedures and structural landmarks in human natural language route instructions (Sections 6.6.2 and 6.7). The representations of the HSSH allow MARCO to represent the simple and complex spatial entities, relationships, and procedures needed to execute route instructions. These experiments show that the lower levels of the HSSH

(Causal and Local Metrical) have a very high impact in following people's natural route instructions, while the higher levels (Local Topological and Global Topological) are less often crucial to following spatial route instructions. However, a Global Metrical map is not needed to follow route instructions.

One result of this research is a software system, MARCO, that parses a route instruction text and follows the described route by executing navigation procedures. MARCO connects the linguistic information in instructions texts with semantic representations – imperative reactive procedures grounded in navigation actions and declarative structured relations grounded in active perception. One application for MARCO is as a human-robot interface for mobile robots, which would be useful in domains such as urban search and rescue, astronaut EVA assistance, in-car navigation systems, and intelligent wheelchairs.

7.0.1 Future Work

MARCO does not yet match human performance, but captures the important spatial representations and procedures for navigating through large-scale spaces. In the changes from Corpus 1, the development corpus, to Corpus 2, only five out of 17 changes altered the spatial or procedural representation; the remainder were linguistic, programming MARCO to recognize different ways of expressing the same concepts.

In the discrepancy analysis (Section 6.9), nearly all the remaining differences in performance between MARCO and people are primarily linguistic. 90 total instructions were analyzed over the three corpora and only 31 had MARCO substantially underperforming people, by > 25 percentage points. Conversely, on 13 route instructions, MARCO succeeded substantially more often. Of the 31 where MARCO underperformed, twelve reflect syntactic parsing or phrase modeling issues (which the representation could handle, but the modeler could not), ten were discourse modeling challenges of combining descriptions across utterances, and three unknown words not matched in WordNet. Only four were spatial or procedural; most where people were able to recover from errors in the

instructions, but MARCO was not. Much work remains on the linguistic front; less work remains in modeling basic large-scale spatial relations and procedures.

The MARCO architecture presented here is a snapshot of an iterative development process, not a turn-key system. Though MARCO is currently capable of following most natural instructions in the development and test corpora as well as people do, some issues remain. First, some unmet challenges remain in the initial development and testing corpora, especially in the worse-rated instructions. Additional methods are necessary if the developer wishes to equal human performance on all instructions, no matter how flawed or how rare. MARCO would be more capable if extended with bootstrap learning; planning, acting, and perceiving under uncertainty; interactive dialogue; and modeling other kinds of actions and relations, especially in small-scale space.

We have shown the initial and post-analysis performance on a novel language and task corpus, documenting the changes necessary for MARCO to approach human performance on these new instructions. While novel challenges will become less frequent as the development (or training) corpus widens, further unknown challenges may lurk in unseen instructions from other directors or for related tasks.

Future work could apply the findings of this dissertation – how people produce and follow route instructions – to the task of generating route instructions. Route instruction generation systems, such as map kiosks, web, phone, and in-car services, can be improved. By recognizing and carefully describing the more difficult segments in route instructions, a system might generate route instruction texts which are more natural, more easy to follow, and more reliable. Another application could check route instruction texts from other sources for completeness, clarity, coherence, conciseness, and likelihood of success. Additionally, the system could produce instructions matching the follower's preferred style.

7.1 Empirical examination of route instruction following

In the beginning of this dissertation, we proposed five questions that must be answered to build an instruction-following system for any domain. We now can answer these questions for spatial route instructions and suggest features which will generalize to other domains.

To discover how people describe spatial routes in navigation instructions, we gathered a corpus of over 1500 route instructions from 30 directors. The instruction elicitation procedure was designed such that directors produce natural instructions, including natural errors and omissions, as in a naturalistic setting. Specifically, the directors are recalling an environment each learned through first-person experience, planning the route from memory, and writing free-form route instructions without limit on grammar, vocabulary, or style.

To measure the quality of these instructions and get a performance standard, we gave the instructions to 100 other people to follow. These human followers had to rely in the information to navigate, because they did not have experience in these environments outside of following the instructions. The route instructions from our directors varied substantially in style, subjective quality, and success rate. People following the instructions reached the intended destination on 25% to 95% of the trials, on the lowest- and highest-rated route instructions respectively.

The major conceptual contribution of the MARCO architecture is deferred handling of ambiguity, both for referring phrase resolution and for modeling instructions as a sequence of procedures. Referring phrases vastly under-specify the configuration, even combined with other knowledge explicit in the instructions or in unspoken shared common heuristics – common-sense. Instructions likewise under-specify the sequence of procedures a follower must execute to accomplish the described task. The performance of MARCO on natural instructions shows the robustness and sufficiency of letting the environment resolve both referring phrase and procedural ambiguity.

By selectively removing language, action, perception, and spatial reasoning

abilities, the evaluation measured the importance of each ability for following spatial route instructions. These experiments reveal the importance of various skills for correctly following instructions. Even where MARCO does not equal human performance, the relative drop-off in performance, the impact, reveals the role of the representation, behavior, or heuristic in instruction following. From an engineering point of view, this is useful to build systems to follow instructions, especially concentrating on high impact components. From a scientific point of view, this is important as a computational model of route instruction following at the level of information processing. The performance of the model exposes individual differences between and within directors at the deep level of what is required to follow the instructions, rather than what is mentioned in passing.

Appendix A

Human Experiment Materials

A.1 Software configuration

To run the experiment, you need a licensed copy of WorldViz Vizard Virtual Reality software Vizard (2006) installed. These experiments were run with Vizard 2.53g and Python 2.4 Python (2007). You will also need to install the Tkinter widget kit (Lundh, 1999).

Next, check the code out of the cvs repository:

```
stankiewicz@lab.psy.utexas.edu:/Volumes/LabData/LabUsers/cvs
```

Check out the `DirectionVizard` module. This is an alias that will check out the experiment control scripts, the shared libraries (`NavigationModules`) and media (`SharedMedia`), and the subject information spreadsheet (`SubjectInfo`).

Gender	Env	Pos Set	Motion
<i>Experiment 2</i>			
Male	Grid	0	Discrete
Male	Grid	1	Discrete
Male	Jelly	0	Discrete
Male	Jelly	1	Discrete
Male	L	0	Discrete
Male	L	1	Discrete
Female	Grid	0	Discrete
Female	Grid	1	Discrete
Female	Jelly	0	Discrete
Female	Jelly	1	Discrete
Female	L	0	Discrete
Female	L	1	Discrete
<i>Experiment 3</i>			
	Male	Grid	0
Continuous			
Male	Grid	1	Continuous
Male	Jelly	0	Continuous
Male	Jelly	1	Continuous
Male	L	0	Continuous
Male	L	1	Continuous
Female	Grid	0	Continuous
Female	Grid	1	Continuous
Female	Jelly	0	Continuous
Female	Jelly	1	Continuous
Female	L	0	Continuous
Female	L	1	Continuous

Table A.1: Combinations for director experiments 2 (Corpus 2) and 3 (Corpus 3). ‘Grid’ refers the *Compact* map, ‘L’ to the *Medium* map, and ‘Jelly’ to the *Sparse* map.

A.2 Running the experiment

First, get consent, add the subject's demographic information to `SubjectInfo/SubjectInfo.xls`, and brief the subject with the appropriate instructions.

Run `Directions.py` in Vizard. You should see this option screen in Figure A.1.

To run a director, enter the subject's ID and the environment name in the 'Message' field, as I've done with `MTM Grid`. To use the alternate set of position names ("Position Set 1"), select the `HMD` option.

The `SubjectID` is the subjects initials, with `X` as a middle initial if the subject has no middle name. If the `SubjectID` would be the same as another subject's ID, change the middle initial.

The codes for the three environments are `Grid`, `Jelly`, and `L`.

To use the joystick or arrow keys for Continuous Movement, enter the letter 'C' after the environment code in the Message field. If you enter 'D' or no code, the experiment will use the Discrete motion code.

Allow the director to get acclimatized to the interface in one environment, then press the 'escape key' to exit the experiment, and restart the experiment in the trial environment.

The `Option 2` check box can be used to skip the training phase for the director, during acclimatization or if you need to restart the experiment for the director.

To run a follower, select `Option 1` and enter into the Message field (1) the subject's id, (2) the string `All` for the environment name, and (3) the run number, for instance `MTM All 2`. To use the alternate set of position names, select the `HMD` option. This will give the follower one set of instructions for each route in all three environments, in a carefully specified order.

If the `Tracker` is clicked, Vizard will save a snapshot of each view while navigating.

After the subject finishes running, add their logs (in `SubjectLogs/`) to `cvs` and check them in.

A.3 Example Consent Form

Navigating through Complex Environments

You are invited to participate in a study of human navigation. This study is part of an ongoing research program in the area of human spatial navigation carried out by Brian Stankiewicz, Ph.D. Participation in this study is completely voluntary. If you decide to participate, you will be one of approximately fifty people in the study. If you decide to participate, the total length of the experiment is approximately one hour. For your participation you will be receive course credit.

This experiment is a study in human spatial navigation and involves learning the spatial layout and traveling though environments that will be displayed on a computer screen. The study's purpose is to inform us about how humans navigate through complex environments. The risks in the study are no more than those encountered while working on a computer. If you are experiencing any physical discomfort please inform the experimenter immediately and the experimenter will try and alleviate any discomfort.

Any information that is obtained in connection with this study and that can be identified with you, will remain confidential and will be disclosed only with your permission. Often times in this line of research, subject's initials are used to differentiate between different subjects in a research paper. Your name will not be associated with these initials in any paper or presentation.

Your decision to participate or to decide not to participate will not affect your present or future relationship with The University of Texas at Austin.

If you have any questions about the study, please ask me. If you have any questions later, you may call Professor Brian Stankiewicz at 512-232-9373. If you have any questions or concerns about your treatment as a research participant in this study, all Professor Lisa Leiden, Ph.D., Chair of The University of Texas at Austin Institutional Review Board for the Protection of Human Subjects, 512-471-8871 or email: orsc@uts.cc.utexas.edu.

You will be given a copy of this consent form for your records.

You are making a decision whether or not to participate. Your signature below indicates that you have read the information provided above and have decided to participate in the study. If you later decide that you do not want to participate in the study, simply tell me. You may discontinue your participation in this study at any time.

Printed Name of Participant

Signature of Participant

Date

Signature of Investigator

Date

A.4 Director Guide

This study investigates how people give route instructions in an indoor environment. The study will be conducted in an unfamiliar virtual building. During the experiment you will participate in three basic tasks: (1) Exploration, (2) Navigation Quiz, and (3) Giving route instructions.

A.4.1 Exploration

Before giving instructions you will explore a virtual building.

You will move through the building by making key presses. Using the number pad, you will move forward by pressing the '8' key, rotate right by pressing the '6' key and rotate left by pressing the '4' key.

As you navigate through the building you will hear a series of 'target locations', specified by a voice announcing the position number when you walk over the position. For example, you might hear "Position 2" when you walk into an intersection. Your goal in this task is to learn the building well enough to navigate and give route instructions between these target locations.

You will have 120 forward movements to learn the building.

A.4.2 Navigation Quiz

After the building exploration phase, we will quiz how well you can navigate through the environment. You will be placed at one of the target positions. Press the space bar to show the environment. You can turn around to orient yourself, but you cannot move forward yet. When you are ready to move, press the '0' (zero) key on the numeric keypad.

The computer will then instruct you to move to a particular target location, e.g., "Go to Position 4." Your task is to move to that target location taking the shortest distance path. The announcement sound for the target location is turned off. When you have done your best to reach the goal, press the space bar.

If you were not at the correct target, the screen will turn red. If you reached the target location by a longer than necessary path, the screen will turn blue. If you go to the target location by taking a short path, the screen will turn green. You will then be transported to a new location for another trial. Press the space bar again to show the environment.

When you have shown you can efficiently navigate through the environment (i.e., take the shortest path for seven straight test trials), you will move on to the next phase of the experiment. If after 25 trials, you still need more experience, you will participate in another Exploration phase followed by another Navigation Quiz.

A.4.3 Giving Route Instructions

After you've passed the Navigation Quiz, you will be asked to give route instructions. As in the Navigation Quiz, you will be placed at one of the target positions and be allowed to turn around to orient yourself. Press the space bar again to show the environment.

The computer will tell you which target location you are current at in the environment (e.g., "Position 2") and tell you to "Turn around," so you can see the position. When you have re-oriented yourself and are ready to give instructions press the '0' key on the numeric keypad. If you recognize or remember the position, you do not need to turn around, and can press '0' immediately. Until you press '0', you can turn but not move forward.

A text entry window will appear asking for route instructions from the current location to another target location, e.g., "Enter directions to get from Position 2 to Position 6." These instructions will be given to another subject who does not know this environment, but has experience in a practice environment. Your task is to write instructions that will reliably direct this subject from the current position to the specified target location.

Use complete sentences and end them with periods. Each set of directions should stand alone. The followers will receive your and others' directions in a mixed order. Do your best if you cannot give complete or exact instructions.

The other people following your instructions will be placed at the starting position facing any one of the four directions. The direction the follower will start facing is not related to the directions you face while orienting yourself at the location.

After finishing the instructions, you will be asked to move to that target location and press the space bar when you are done. After moving, you will answer two questions: (1) How certain are you that you've reached the target location? (2) How good do you think your instructions were? You will alternate between giving instructions, navigating through the environment, and rating your performance until all positions have been used.

To familiarize you with the experiment you will run through a practice environment. Use this time to become comfortable with how to move through the environment and how to give instructions. If you have any questions, please ask at any time.

A.5 Director Key Meanings

8 Move Forward

4 Turn Left

6 Turn Right

0 Done with Turning Around

space Show Environment / At Destination

t Repeat the current target position

w which hotspot

A.6 Instruction Follower Guide

This study investigates how people follow route instructions in an indoor environment. We are interested in finding out what makes a good set of route instructions and how people follow instructions, even when the instructions contain mistakes. The study will be conducted in a virtual building. During the experiment you will read and follow sets of route instructions that guide you through the building.

A.6.1 Movement Controls

You will move through the building by making key presses. Using the number pad, you will move forward by pressing the '8' key, rotate right by pressing the '6' key and rotate left by pressing the '4' key.

A.6.2 Following Route Instructions

You will follow sets of route instructions that someone else gave. These instructions vary in how and how well they describe the route and destination, so just try to follow each to the best of your ability. The computer will display a set of instructions on the computer screen in a popup window. Read through the instructions and click OK when you are ready. Then, you will be placed in the environment at the starting position.

Some of the instructions may have mistakes or even be blank. Try your best to get to the destination, even if the instructions are incomplete or there are small mistakes (for instance, it says "left" when there is only a right). You may not be facing the same direction as the direction giver was when the instructions were written. You can reread the instructions while you are moving by pressing the 'd' (instructions) key. When you have done your best to navigate to the end position, press the space bar.

After moving, you will answer two questions on popup windows: (1) How certain are you that you've reached the target location? (2) How good do you think the instructions were? After you've answered both questions, the screen will be blank again. When you are

ready for the next set of instructions, press the space bar again. You can take a break any time you like while the screen is blank between each direction set.

To familiarize you with the experiment you will be run in a set of “practice” sessions. You will be following instructions in a small environment. The purpose of these practice sessions is to get you familiar with how to move through the environment and how to give and follow instructions. If you have any questions, please ask at any time. Once you are comfortable with the experiment, we will start the full set, which is about 120 sets of instructions.

A.7 Follower Key Meanings

8 Move Forward

4 Turn Left

6 Turn Right

space Next Directions / At Destination

d Reshow the **d**irections while moving

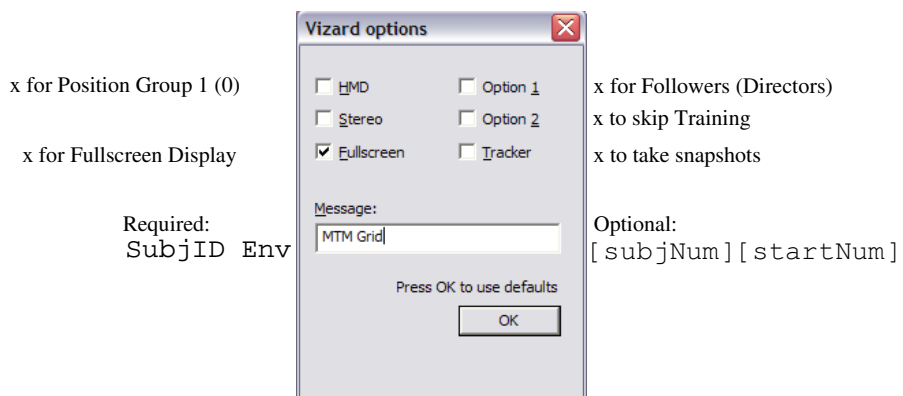


Figure A.1: Vizard Experiment Starting Dialog

Appendix B

MARCO Ablation Options

This section describes the ablation options used to configure MARCO. Most options are Boolean on/off switches. This section notes the name of each option, a description of it, examples of usage scenarios, and any fallback default behavior.

B.1 Options: Fundamentals

Fundamental abilities: Skills that are explicitly used in instructions, e.g. counting, traveling until.

Distance Count *Impact: 48%* Move to distance count. Move forward to the second alley. Move forward once.

Face Description *Impact: 75%* Turn until a view description is met. Face the chair. With your back to the blank wall,

Perspective Taking *Impact: 67%* Project a different perspective from current view. to your left is a chair. Go to the right of the stand

Travel Until *Impact: 59%* Travel until the described view. Go to the end of this hallway. When you get to the red brick hall, ...

Turn Direction *Impact: 15%* Turn towards a specified direction. Turn left. Take a right.

Use Find *Impact: 17%* Use the find behavior. it's at the corner of the yellow floor and the grassy floor. Find the blue road.

Use Follow *Impact: 2%* Use the follow behavior. Follow the alley around go all the way down the winding hall

View Memory *Impact: 81%* Remember views even after turning or traveling. go down toward the longer end of the hallway. Going away from the coat hanger

B.2 Options: Conditionals

Conditional abilities: Skills that are explicit in instructions, but with conditionals.

Declare Goal Cond *Impact: 18%* Move to satisfy condition on DeclareGoal. when you come to the intersection with ..., you are at position 6 once in the L you're in position 5.

Distance Before *Impact: 2%* Move to distance targets of X before Y. Stop at last intersection before the bench one intersection before the coat rack will be 3

Distance Past *Impact: 3%* Move to distance targets of X past Y. One segment past the chair is Position 3. ...until two sections past the lamp

Face Away *Impact: 1%* Face the away condition of Turn. Go away from the clothes hanger. Face away from the dead end.

Face Explicit *Impact: 11%* Face the explicit argument of Face. Face the wall face the easel and dark gray stone floor.

Face Onto *Impact: 3%* Face the onto condition of Turn. take a left onto the blue path. turn right onto the stone

Face Purpose *Impact: 7%* Face the purpose condition of Turn. turn to face the long red hallway. turn left so that you face another bench

Face Toward *Impact: 1%* Face the toward condition of Travel. go towards the red brick floor. move toward the bench.

Face Travel Args *Impact: 8%* Face Travel arguments. take the yellow path to the wood path intersection Go down the short hall

Face View *Impact: 3%* Face a standalone description of a view. you should have butterfly pictures in front of you you see a little bit of yellow on the floor

Stop Cond *Impact: 4%* Interpret 'stop when COND' as 'go until COND.' stop at the first intersection stop when the first fish hallway to your left appears

Travel Location *Impact: 1%* Move to syntactically marked precondition location of a Travel. at the wood path intersection, take the wood path from the easel, move one block

Travel Past *Impact: 1%* Travel past conditions. move one place past the brick floor tiling pass the easle that is sitting in the hall

Travel Precond *Impact: 8%* Turn to syntactically marked precondition of a Travel. facing the easel, go forward. with your back to the wall move along the green ...

Turn Location *Impact: 7%* Move to syntactically marked precondition location of a Turn. once you hit the dining chair, turn right At the brick

hallway, turn towards ...

Turn Postcond *Impact: 2%* Travel to syntactically marked postcondition pose of Turn. take a right to the end of the hall take a right onto the green path all the way to the end of the hall

Turn Precond *Impact: 2%* Turn to precondition precondition pose of a Turn. With your back facing the wall turn right facing the long aisle turn left

B.3 Options: Heuristics

Heuristics Hints: Strategies that enforce simple implicit inferences, e.g. face an open path before traveling.

Face Distance *Impact: 4%* Face at least one of the distance units (e.g. intersections, streets, movements) of Travels.

Face Past *Impact: 2%* Face objects the Travel will pass.

Face Until *Impact: 33%* Face the termination condition of Travels.

Face Until Post Dist *Impact: 2%* Re-face the termination condition of Travels after going distance, to catch over-estimates.

Look Ahead For Travel Term *Impact: 6%* Look ahead in instructions for termination for travel actions.

Look Ahead For Travel Term Desc *Impact: 4%* Look ahead in instructions for termination for travel actions in descriptions. Go foward. Look for butterflies
Walk towards the brick hallway. At one end of the brick hallway,
there is a chair.

Look Ahead For Travel Term Loc *Impact: 2%* Look ahead in instructions for termination for travel actions in location phrases. go towards the easle but stop at the closest concrete square closest to the easletake a left onto the pink path. at the next intersection, ...

Propagate Context Info *Impact: 1%* Propagate the context information to embedded compound action specifications.

Reverse Turn *Impact: 3%* If last action was a turn, but now facing a wall, turn around.

Travel Between Turns *Impact: 1%* Make a travel between consecutive turns. take a left then a right. left. left.

Travel On Final Turn *Impact: 4%* Make a final travel forward when the last action was a turn. with your back to the wall turn left and move one block. turn right. go down the butterfly walled/blue floored hallway. make a left at the hatrack.

Travel On Final View *Impact: 3%* Make a final travel forward when the last action was a view. turn left and you should see a barstool. this is position 7. You should be able to see the grassy hall to your right. This is Position 1.

Travel To Next *Impact: 5%* Travel to next match when last action was turn. ...take a left. Go down to the corner. go until hall ends. take a left. go until hall ends.

Turn Between Travels *Impact: 1%* Make a turn between consecutive travels. Go forward down the hall until a hall opens to your left. Go forward one segment. yellow hall to wooden hall. then one space forward.

Turn Explicit *Impact: 1%* Turn towards an explicitly mentioned direction before checking condition.

Turn Post Reset Cache *Impact: 1%* Reset anaphora cache after interpreting turn.

Turn Pre Reset Cache *Impact: 1%* Reset anaphora cache before interpreting turn.

Turn Term Reset Cache *Impact: 1%* Reset anaphora cache before interpreting turn termination.

Turn Toward Path *Impact: 2%* If no explicit turn direction, turn towards a visible path instead of a wall.

B.4 Options: Recoveries

Error recovery: Strategies that attempt to recover from a failed action.

Check After Turn Find *Impact: 1%* Check the until condition of the find between the turn and travel.

Find Face *Impact: 16%* Find when Face does not find a satisfying pose visible from this place.

Find Face Travel *Impact: 2%* Use Find as fallback for Face even if not ImplicitTravel

Travel To Distant View *Impact: 2%* Travel when Facing a satisfying pose distantly visible from this place. at the next corner, take a right at the lamp. go forward. then left all the way down the blue hall

B.5 Options: Tweaks

Tweaks: parameters that have small effects on how a behavior is executed.

Declare Goal For Position *Impact: 1%* DeclareGoal whenever a position is mentioned.

Travel Empty *Impact: 1%* Travel forward without arguments.

Travel No Termination *Impact: 2%* Travel forward without termination conditions.

B.6 Options: Linguistics

Linguistic parameters: parameters that affect the interpretation of text.

Declare Goal Idiom *Impact: 8%* Treat 'This is (it|Position X) as idiom. this is 6.

Position 6 is the next intersection as you follow the red-brick hallway.

Face Declaratives *Impact: 26%* Enforce declaratives with Face statements. to your right you should see an alley with grey carpet. there is a bench there.

Fuzzy Meanings *Impact: 7%* Use broader definitions of concepts. chair gray

Raw Reference Resolution *Impact: 4%* Fill a reference phrase or pronoun with the corresponding noun phrase, even if not limited by syntax. Face toward the hatrack. Walk the one segment to it. there should be blue carpet on the first alley. walk to that and turn right.

Recognize Across *Impact: 3%* Recognize across phrases. move across one yellow panel. turn right across 2 black stone floors.

Recognize Arrive Frame *Impact: 2%* Handle arrive frames as travel until. turn right when you reach the end. Once you get to the floral carpeted hallway, look for the easle.

Recognize Complex Expressions *Impact: 6%* Recognize complex expressions. Component options: Recognize Fictive Turn Intersections, Recognize Struct Frame, Recognize Pass Frame, Recognize Dir Turn, Recognize Distal Determiners, Recognize Last, Recognize Negative, Recognize Negative Compound, Recognize Standalone Arrive, Recognize Struct Agent Frame, Recognize Until Loc Dist, Recognize Until View

Recognize Count *Impact: 5%* Recognize counts in noun phrases. walk past two chairs and to the lamp. This empty intersection is Position 7.

Recognize Dir Turn *Impact: 3%* Handle e.g.'(until) the last right.' there should be only once choice, to turn right. this left turn should have a yellow floor.

Recognize Distal Determiners *Impact: 2%* Recognize determiners like 'that' and 'other' as marking distant entities. go down the red hall until you see the blue hall. at that intersection stop. then turn left until you see another bench and move to it.

Recognize Fictive Turn Intersections *Impact: 2%* Handle 'where you (can) turn/travel to the right...' move forward until you can turn left again. go straight until you can either go left or towards a dead end.

Recognize Last *Impact: 4%* Recognize last as order adj. until you come to the last empty intersection before the easle Follow this down until you come to the second to last left.

Recognize Negative *Impact: 0%* Recognize negative phrases. when there is not a wall to your left, go straight. ...toward the pictures of not-butterflies

Recognize Negative Compound *Impact: 1%* Recognize negative compounds phrases. not a bench or a stool.

Recognize Noun Noun Compound *Impact: 2%* Recognize noun noun-compounds. down the long butterfly hallway, with blue walls. Face the pink-flowered carpet hall

Recognize Pathdir *Impact: 2%* Handle 'Only one way to go.' only one way to go face the direction with the easel.

Recognize Standalone Arrive *Impact: 2%* Handle sentences like 'You will hit an l. the very first section you come to, you will be at 5 you will hit an intersection with black stone floors.'

Recognize Struct Agent Frame *Impact: 2%* Handle sentences like 'You will have to take a right. ' You will have to take a right onto a floor that is black.'

Recognize Struct Frame *Impact: 2%* Handle frames with structural descriptions as verbs, like 'where the paths cross.'. when road ends, go right. then a left down the red hall until it intersects with the rose hall

Recognize Structural *Impact: 3%* Handle structural adjectives, especially long and short. turn until you face the short end of the hallway with blue flooring. go down the longer part of the hall with blue rectangles.

Recognize Take Turn Frame *Impact: 10%* Handle 'Take ((the nth)l) (rightleft) [turn].'
make a left. take your second right.

Recognize Until Loc Dist *Impact: 2%* Handle Travel UNTIL LOC DIST.

Recognize Until View *Impact: 2%* Handle Turn until VIEW. turn until you face the hallway with the green floor from one turn until you see a corner of blue carpet in a side alley.

Reference Resolution *Impact: 2%* Fill a reference phrase or pronoun with the corresponding noun phrase. you should see an alley to your left. take it. you will be looking for a pink flowered path. When you reach this path, ...

Spellcheck *Impact: 2%* Do spellchecking and replace unknown words with known.

B.7 Options: Implicits

Implicit Actions *Impact: 61%* Infer and perform implicit procedures. Component options: Implicit Travel, Implicit Turn

Implicit Exploration *Impact: 22%* Active resolution of referring phrases. Component options: Travel To Distant View, Find Face, Recognize Structural

Implicit Pragmatic *Impact: 15%* Discourse and idiomatic cues. Component options: Declare Goal Idiom, Recognize Pathdir, Look Ahead For Travel Term, Travel On Final Turn, Travel Between Turns, Turn Between Travels, Travel To Next, Travel On Final View, Propagate Context Info

Implicit Pragmatic Cross Utterance *Impact: 9%* Pragmatics across utterances. Component options: Look Ahead For Travel Term, Travel On Final Turn, Travel Between Turns, Turn Between Travels, Travel To Next, Travel On Final View, Propagate Context Info

Implicit Pragmatic Per Utterance *Impact: 8%* Pragmatics per utterance. Component options: Declare Goal Idiom, Recognize Pathdir

Implicit Semantic *Impact: 51%* Complex action frames. Component options: Recognize Take Turn Frame, Recognize Arrive Frame, Face Travel Args, Face Distance, Face Until, Face Past, Turn Toward Path

Implicit Syntactic *Impact: 37%* Prepositional phrase markings. Component options: Face Purpose, Travel Precond, Turn Precond, Turn Postcond, Declare Goal Cond, Turn Location, Travel Location, Stop Cond, Declare Goal For Position

Implicit Travel *Impact: 36%* Infer and perform implicit travels.

Implicit Turn *Impact: 50%* Infer and perform implicit turns.

B.8 Options: Landmarks

Recognize different sorts of landmarks.

Appearance Landmarks *Impact: 35%* Recognizing simple perceptual attributes, e.g. color, texture. take one movement towards the blue corridor Go to the hallway that has the blue tiles and the orange butterflies on the wall.

Causal Landmarks *Impact: 47%* Recognizing simple structural landmarks, e.g. paths, walls, positions. with your back to the wall turn right. take a left onto the black path

Intersection Landmarks *Impact: 25%* Recognizing intersection structural landmarks, e.g. dead ends, T intersections, corners. the dead end is position 4. at the first intersection after the lamp,

Object Landmarks *Impact: 24%* Recognizing distinct objects, e.g. furniture and pictures. The intersection with the chair is Position 4. Go towards the coat rack and take a left at the coat rack.

Structural Landmarks *Impact: 58%* Recognizing simple structural landmarks. Component options: Causal Landmarks, Intersection Landmarks

B.9 Options: HSSH

Divide methods by what level of the SSH they require.

Causal *Impact: 98%* All Causal actions. Component options: Distance Count, Turn Direction, Travel Until, Face Description, Object Landmarks, Causal Landmarks

Closed Loop Causal *Impact: 89%* Closed loop causal control laws. Component options: Travel Until, Face Description, Object Landmarks

Local Metrical *Impact: 76%* Local Metrical. Component options: Travel To Distant View, View Memory, Perspective Taking, Face Distance

Local Topological *Impact: 55%* Local Topological. Component options: Recognize Structural, Intersection Landmarks, Turn Toward Path, Reverse Turn, Look Ahead For Travel Term, Recognize Count, Recognize Dir Turn, Recognize Pathdir, Face Until, Face Until Post Dist

Open Loop Causal *Impact: 54%* Open loop causal control laws. Component options: Distance Count, Turn Direction

Topological *Impact: 25%* Topological. Component options: Use Follow, Use Find, Travel To Next, Travel Between Turns, Turn Between Travels, Travel On Final Turn

B.10 Options: Comparison

Comparisons to different agents or development baselines.

Corpus2 Options *Impact: 8%* Options added when examining corpus 2. Component options: Declare Goal For Position, Propagate Context Info, Raw Reference Resolution, Recognize Across, Recognize Count, Recognize Dir Turn, Recognize Distal Determiners, Recognize Last, Recognize Negative, Recognize Negative Compound, Recognize Standalone Arrive, Recognize Struct Agent Frame, Recognize Until Loc Dist, Recognize Until View, Reverse Turn, Spellcheck, Travel On Final View

IBL Options *Impact: 4%* Options that distinguish Marco from the IBL agent. Component options: Reverse Turn, Turn Toward Path, Face Distance, Implicit Pragmatic

Appendix C

Glossary and Language Model

C.1 Glossary

C.1.1 Route Instruction Analysis Vocabulary

Route Instruction Units

Utterance Sentence or fragment of natural language text.

Command An explicit imperative utterance.

Instructions Linguistic or pictorial description of a procedure of explicit commands and descriptive utterances, describing what the follower agent should do.

Route Instruction Navigation task-specific description of where and how the follower agent should move.

Procedural Specification A representation of the constraints that route instructions place on a reactive procedure, consisting of the parameters, constraints, and conditions governing the activation of discrete Causal actions. Equivalent to *tasks* at the sequencing layer of multi-tiered architectures. Task-dependent representation of how the follower agent achieves an instruction, grounded in the follower's capabilities.

C.1.2 Abbreviations

POMDP Partially Observable Markov Decision Process

SSH Spatial Semantic Hierarchy

HSSH Hybrid Spatial Semantic Hierarchy

VRML Virtual Reality Markup Language

C.2 Route Instruction Grammar

C.2.1 Verbs

TURN Change in orientation or switch in paths: turn, take (a left), make (a right turn), go (left)

ORIENT Change in rotation relative to an external frame of reference. face, put (your back against the wall), stand (so that ...)

TRAVEL Move between places: walk, go, move, follow, take (the path)

STOP Cease motion: stop

PASS Move past an entity: pass

ARRIVE Move until a description: get to, meet, come to, hit, enter

FIND Undirected search: find, look for, go (to where ...)

IS Be verb: is, match, are

HAS Possession verb: contain, has

SEE Local observation verb: see, look

LOC Location verb: standing, located

STRUCT Describe environment structure by movement metaphor: intersect, meet, brings, hits, runs into

C.2.2 Nouns

PLACE Place name or description. position 2, a spot with a chair

PATH Way to travel between places. hallway, road, path

STRUCT Local place topology. end of the hall, corner, 'T' intersection

REGION Large-scale space region. the Eiffel Tower area

DIST_UNIT Units of distance. segments, movements, times, PATH, STRUCT

SIDE A direction or directions. the sides, direction, way, the back

OBJ Discrete localized object. furniture, wall, picture

AGENT Agent that can act in the world. you, yourself, the follower

VIEW Description of view. to <where you see a chair>, from <this perspective>

REF Reference: it, that, there

VIEWDESC Description of the expected view: PATH | OBJ | STRUCT | REGION | VIEW |
DESC | REF

C.2.3 Turn Command Arguments

Phrase	Arg.	Action Model
<i>Pre</i>		
COND	DESC	Face ^p (faced:Desc)
LOC	VIEWD	Travel ^p (until:ViewD)

While

DIR	DIR	Turn ^p (direction:Dir)
"	SIDE	Turn ^p (direction:Dir)

Post

PURPOSE	DESC	Face ^p (faced:Desc)
TOWARD	VIEWD	Face ^p (faced:ViewD)
AWAY	VIEWD	Face ^p (faced:ViewD)
ONTO	PATH	Face ^p (faced:Path)

COND After this condition holds, turn evaluate other arguments: <with your back to the wall>; <facing the blue path>

LOC Where to turn: left <at the coat rack>, take a right <two intersections after the lamp>

DIR Turn direction relative to agent: <left>, <right>, <around>

SIDE Facing so that the referent is on the named side: <to your left>

PURPOSE Turn so that this condition holds: place <your back to the wall>; turn <so that the wall is on your left>; turn <to locate the easel>

TOWARD Facing the referent: face <the hat rack>, <toward the longer path>

AWAY Facing away from the referent: away from the rose path intersection

ONTO Turn onto this path: <into the yellow tiled hall>, <onto the red brick>

C.2.4 Travel Command Arguments

Phrase	Arg.	Action Model
<i>Pre</i>		
ALONG	PATH	Face ^p (faced:ViewD)
COND	DESC	Face ^p (faced:Desc)
LOC	VIEWD	Travel ^p (until:ViewD)
TOWARD	VIEWD	Face ^p (faced:ViewD)
AWAY	VIEWD	Face ^p (faced:ViewD)
DIR	DIR	Turn ^p (direction:Dir)
"	SIDE	Turn ^p (direction:Dir)
<i>While</i>		
PAST	VIEWD	Travel ^p (past:ViewD)
DIST	DIST	Travel ^p (dist:Dist)
<i>Post</i>		
UNTIL	VIEWD	Travel ^p (until:ViewD)

ALONG Travel on this path: along the grass carpet, down the hall with the yellow stone floors, take it

TOWARD Direction on a path toward an allocentric entity: toward the hat rack, in the direction of the red floor

AWAY Direction on a path to face the referent: away from the bench

DIR Travel Direction on a path relative to agent: straight, left, right

PAST Move past entity on this path, in the direction of the referent but further than it: past
the stool, passing the lamp, beyond that

QUAL-DIST Qualitative description of distance: all of the way down, far, very
end

QUANT-DIST COUNT (DIST-UNIT) Quantitative description of distance: one movement,
two alleys away from, about five intersections, first left

UNTIL Continue travel until condition: stop at the easel, to the lamp, until
you reach the pink walkway

C.2.5 Description Utterance Arguments

DESC

- LOC* (there | where | COND)* OBJ | VIEW | STRUCT | LOC | POSITION V/Is
(Appear | Side | Loc)
- (there | where) V/Is OBJ | VIEW | STRUCT | LOC
- PATH V/STRUCT PATH
- (PATH | STRUCT | LOC | REF) V/HAS (PATH | STRUCT | LOC | REF)
- LOC* ARRIVE

COND Conditional for (ARRIVE | VIEW | ORIENT | LOC | DESC) when ..., once
..., if ...

ARRIVE Description of termination of motion you get to the ..., you come
to an intersection with ...

C.2.6 Adjectives

APPEAR Appearance, floor color and texture, pictures: blue, green octagon, fish
on the wall, wooden floored

STRUCTURAL Length and shape of path: long, short

STRUCT-TYPE Local topology and geometry of environment structure: four-way, 'T'

C.2.7 Adjectival Phrases

ON : P/ON PATH Entity is on this path: in this hall, on the long path

BETWEEN : P/BETWEEN VIEWDESC VIEWDESC Entity is between two other entities:
between the chair and the hatrack, between the two intersections
containing furniture

SIDE : P/SIDE DIR VIEWDESC? Location of the object relative to another entity in
front of the easel, to your left, behind you, on one side

AGAINST : P/AGAINST VIEWDESC Object is in contact with or adjacent to the referent.
to the wall, against the chair

LOC : P/LOC VIEWDESC Location of object: by the coat rack, in the corner,
at the end of the hall

DETAIL : (P/DETAIL | V/HAS) OBJ Descriptive detail: with the blue floor,
containing an easel

PART : P/PART OBJ Composition of object: end <of the hall>, intersection
<of this and the blue floor>

C.3 Interfaces

C.3.1 Simulation

Inputs and outputs

Agent Actions : travel^a, turn-left^a, turn-right^a, declare-goal^a

Test Actions : teleport^a, set-goal^a

Observations : A variable-length list of tuples, representing the view to the perceptual horizon (end of the hall). See description below.

Observation Type

The observation is a discrete symbolic representation of each component of the view visible to the end of the hallway in front. The list is composed of an arbitrary number of 6 item tuples, $\langle left, middle, right, f-left, front, f-right \rangle$. This represents the view of a place, its peripheral hallways, and the floor and walls of the hallway segment immediately in front.

Positions *left*, *right*, and *front* describe the floor texture, if any, and are drawn from the set { *Wall*, *Rose*, *Wood*, *Grass*, *Cement*, *BlueTile*, *Brick*, *Stone*, *Honeycomb* }. Positions *f-left* and *f-right* describe the corridor walls, if visible, and are drawn from the set { *End*, *Butterfly*, *Fish*, *Eiffel* }. Position *front* describes any object visible within the intersection and is drawn from the set { *Chair*, *Sofa*, *Barstool*, *Hatrack*, *Easel*, *Lamp*, *Empty* }.

See Figures 6.1 and 6.2 for examples.

C.4 Representation of Procedural Specifications

See Section 5.4.2 for more explanation.

Common parameters

precondition Pre-condition to achieve prior to executing procedure

postcondition Post-condition to achieve while executing procedure

location Where to perform the procedure. A locative pre-condition.

Procedure-specific parameters

Verify^p

Description View description to verify.

Travel^p

along Description of the path to travel along. A topological pre- / during-condition.

distance Estimate of distance to travel. May be discrete (2nd left), continuous (three meters), or qualitative (all the way down). An internal during- / post-condition.

face View description of the view in front of the traveler at the beginning of travel. A perceptual pre-condition.

past View description to achieve during forward motion. A locative during-condition.

until View description of the travel destination. A locative post-condition.

Turn^p

direction Relative direction to turn

Face^p

faced View description to face at the end of the procedure. A perceptual post-condition.

DeclareGoal^p

Goal Name of goal to assert.

C.5 Representation of View Description

See Section 5.4.1 for more explanation.

type Kind of entity. Currently implemented types:

Path Traversal linear structure an agent can travel^a along.

Pathdir Part of a path proceeding in one direction.

Segment Length of path between two places.

Obj Three-dimensional object located at a point, such as furniture, pictures, and walls.

Struct Structural landmark such an intersection or block.

Region Large-scale region of space.

value Token representing an instance of type recognizable to the view description matching code.

dist Distance within the view, if known. Currently a string representing a distance range in the view, e.g. '0' is immediate, '1' is one intersection away, and '1:' is distal.

Relations between entities *Between* ternary relation that the subject is located between two other reference objects, i.e. if the follower is the subject, the reference objects will be on opposite sides of her.

Detail just asserts an unspecified connection between two objects, which may be co-location (e.g. the end with the easel) or part (e.g. the intersection with the brown hallway).

Loc represents that the subject is co-located with (or at or in) the reference object.

On represents that the subject is topologically *on* a reference entity, a path.

Part represents that the entity is part of the other entity, though which is the part may not be apparent from the surface form.

Bibliography

- Agrawala, M. and Stolte, C. (2001). Rendering effective route maps: Improving usability through generalization. In Fiume, E., editor, *ACM SIGGRAPH*, pages 241–250. ACM Press.
- Agre, P. E. and Chapman, D. (1990). What are plans for? *Rob. & Auton. Sys.* , 6:17–34.
- Allen, G. L. (2000). Principles and practices for communicating route knowledge. *Applied Cog. Psych.* , 14(4):333–359.
- Allen, G. L. (2003). Gestures accompanying verbal route directions: Do they point to a new avenue for examining spatial representations? *Spatial Cogn. & Compn.* , 3(4):259–268.
- Allen, J., Ferguson, G., and Stent, A. (2001a). An architecture for more realistic conversational systems. In *Proc. of Intelligent User Intf. 2001 (IUI-01)*, pages 1–8, Sante Fe, NM. ACM Press.
- Allen, J. F., Byron, D. K., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2001b). Toward conversational human-computer interaction. *AI Magazine*, 22(4):27–37.
- Allen, J. F., Miller, B. W., Ringger, E. K., and Sikorski, T. (1996). A robust system for natural spoken dialogue. In *Proc. of 34th Ann. Meeting of the ACL (ACL-96)*, pages 62–70, Santa Cruz, CA. Morgan Kaufmann.

- Alterman, R., Zito-Wolf, R., and Carpenter, T. (1991). Interaction, comprehension, and instruction usage. *J. of the Learning Sciences*, 1(3/4):361–398.
- Anderson, A. (1984). *Semantic and Social Pragmatic Aspects of Meaning in Task-Oriented Dialogue*. PhD thesis, Univ. of Glasgow.
- Anderson, A., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H. S., and Weiniert, R. (1991). The HCRC map task corpus. *Lang. & Speech*, 34(4):351–366.
- André, E., Bosch, G., Herzog, G., and Rist, T. (1986). Coping with the intrinsic and deictic use of spatial prepositions. In *Proc. of AIII: Methodology, Systems, Applications*, pages 375–382, Amsterdam, The Netherlands.
- Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet project. In *Proc. of 36th Ann. Meeting of the ACL (ACL-96)*, Montréal, QB, Canada. Morgan Kaufmann.
- Beeson, P., MacMahon, M., Modayil, J., Murarka, A., Kuipers, B., , and Stankiewicz, B. (2007). Integrating multiple representations of spatial knowledge for mapping, navigation, and communication. In *Proc. of AAAI Spring Symp. on Interaction Challenges for Intelligent Assistants*, Stanford, CA.
- Bindiganavale, R., Schuler, W., Allbeck, J. M., Badler, N. I., Joshi, A. K., and Palmer, M. (2000). Dynamically altering agent behaviors using natural language instructions. In *Proc. of 4th Intl. Conf. on Auton. Agents*, pages 293–300, Barcelona, Spain.
- Bird, S. and Loper, E. (2004). NLTK: The Natural Language Toolkit. In *Proc. of 42nd Ann. Meeting of the ACL (ACL-04)*, Barcelona, Spain.
- Blisard, S. and Skubic, M. (2005). Modeling spatial referencing language for human-robot interaction. In *Proc. of IEEE Intl. Ws. on Robot and Human Interactive Communication (RO-MAN)*, Nashville, TN.

- Blisard, S., Skubic, M., Luke, III, R. H., and Keller, J. M. (2006). 3-D modeling of spatial referencing language for human-robot interaction. In Goodrich et al. (2006), pages 329–330.
- Blocher, A. and Stopp, E. (1998). Time-dependent generation of minimal sets of spatial descriptions. In Olivier and Gapp (1998), pages 57–72.
- Bonnasso, R. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P., and Slack, M. G. (1997). Experiences with an architecture for intelligent, reactive agents. *J. of Exptl. and Theo. AI*, 9(1):237–256.
- Bos, J. (2004). Computational semantics in discourse: Underspecification, resolution, and inference. *J. of Logic, Language and Information*, 13(2):139–157.
- Brown, L. N., Lahar, C. J., and Mosley, J. L. (1998). Age and gender-related differences in strategy use for route information: A map-present direction-giving paradigm. *Env. & Behavior*, 30:123–143.
- Bugmann, G. (2003). Challenges in verbal instruction of domestic robots. In *Proc. of 1st Intl. Ws. on Adv. in Service Rob. (ASER '03)*, pages 112–116, Bardolino, Italy.
- Bugmann, G., Klein, E., Lauria, S., and Kyriacou, T. (2004). Corpus-based robotics: A route instruction example. In *Proc. of Intelligent Auton. Sys.*, pages 96–103, Amsterdam.
- Bugmann, G., Lauria, S., Kyriacou, T., Klein, E., Bos, J., and Coventry, K. (2001). Using verbal instructions for route learning: Instruction analysis. In *Proc. of Towards Intelligent Mobile Robots Conf.*, Manchester, UK.
- Buhl, H. M. (2003). Partner orientation and speaker's knowledge as conflicting parameters in language production. *J. of Psycholinguistic Res.*, 30(6):549–567.
- Burke, J. L., Murphy, R. R., Coovert, M. D., and Riddle, D. L. (2004). Moonlight in Miami: Field study of human-robot interaction in the context of an urban search and

- rescue disaster response training exercise. *Human-Computer Interaction*, 19(1–2):85–116.
- Burnett, G. E. (2000). “Turn right at the traffic lights:” The requirements for landmarks in vehicle navigation systems. *J. of Navigation*, 53(3):499–510.
- Burridge, R. R., Graham, J., Shillcutt, K., Hirsh, R., and Kortenkamp, D. (2003). Experiments with an EVA assistant robot. In *Proc. of 7th Intl. Symp. on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS-03)*, Nara, Japan.
- Byron, D. K. (2002). Resolving pronominal reference to abstract entities. In *Proc. of 40th Ann. Meeting of the ACL (ACL-02)*, pages 80–87, Philadelphia, PA.
- Byron, D. K., Mampilly, T., Sharma, V., and Xu, T. (2005). Utilizing visual attention for cross-modal coreference interpretation. In *Proc. of Context-05*, volume 3554 of *LNCS*, pages 83–96.
- Carletta, J. and Mellish, C. S. (1996). Risk-taking and recovery in task-oriented dialogue. *J. of Pragmatics*, 26(1):71–107.
- Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. (1994). Acting optimally in partially observable stochastic domains. In *Proc. of 12th Natl. Conf. on AI (AAAI-94)*, pages 1023–1028, Seattle, WA. AAAI Press/The MIT Press.
- Chang, N. and Mok, E. (2006). A structured context model for grammar learning. In *Proc. of Intl. JointConf. on Neural Networks*, Vancouver, BC.
- Chang, N., Narayanan, S., and Petruck, M. R. (2002). Putting frames in perspective. In *Proc. of 19th Intl. Conf. on Compl. Ling. (COLING-02)*, Taipei, Taiwan.
- Chapman, D. (1990). *Instruction use in situated activity*. PhD thesis, MIT, Dept. of Elec. Eng. & Comp. Sci., Cambridge, MA. Also Available as MIT, AI Lab. Technical Report 1204.

- Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proc. of 43rd Ann. Meeting of the ACL (ACL-05)*, pages 173–180, Ann Arbor, MI.
- Chewar, C. M. and McCrickard, D. S. (2002). Dynamic route descriptions: Tradeoffs by usage goals and user characteristics. In *Proc. of 2nd Intl. Symp. on Smart Graphics*, pages 71–78. ACM Press.
- Cohen, P. R. (1984). The pragmatics of referring and the modality of communication. *Compl. Intelligence*, 10(2):97–146.
- Cohen, P. R. (1995). *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, MA.
- Cohen, P. R. and Levesque, H. J. (1990a). Persistence, intention, and commitment. In Cohen et al. (1990), pages 33–69.
- Cohen, P. R. and Levesque, H. J. (1990b). Rational interaction as the basis for communication. In Cohen et al. (1990), pages 221–255.
- Cohen, P. R., Morgan, J., and Pollack, M. E., editors (1990). *Intentions in Communication*. MIT Press, Cambridge, MA.
- Cohen, P. R. and Perrault, C. R. (2003). Elements of a plan-based theory of speech acts. In Huget, M.-P., editor, *Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies*, volume 2650 of *LNCS*, pages 1–36. Springer.
- COSIT-99 (1999). *Spatial Information Theory: Cognitive and Computational Foundations of Geog. Info. Sci. (COSIT '99)*, volume 1661 of *LNCS*, Stade, Germany.
- Coventry, K. R. and Garrod, S. C. (2004). *Saying, Seeing, and Acting: The Psychological Semantics of Spatial Prepositions*. Essays in Cognitive Psychology. Psychology Press, Hove and New York.

- Coventry, K. R. and Oliver, P., editors (2002). *Spatial Language : Cognitive and Computational Perspectives*. Kluwer Academic Publishers, Boston, MA.
- Dabbs, J. M., Chang, E.-L., Strong, R. A., and Milun, R. (1988). Spatial ability, navigation strategy, and geographic knowledge among men and women. *Evolution and Human Behavior*, 19:89–98.
- Dale, R., Geldof, S., and Prost, J.-P. (2002). Generating more natural route descriptions. In *2002 Australasian Ws. on Nat. Lang. Proc.* , pages 41–48, Canberra, Australia.
- Dale, R., Geldof, S., and Prost, J.-P. (2003). CORAL: Using natural language generation for navigational assistance. In Oudshoorn, M., editor, *Proc. of 26th Australasian Comp. Sci. Conf. (ACSC2003)*, Adelaide, Australia.
- Dalton, R. C. (2003). The secret is to follow your nose: Route path selection and angularity. *Env. & Behavior*, 35(1):107–131.
- Daniel, M.-P. and Denis, M. (2004). The production of route directions: Investigating conditions that favour conciseness in spatial discourse. *Applied Cog. Psych.* , 18:57–75.
- Daniel, M.-P., Tom, A., Manghi, E., and Denis, M. (2003). Testing the value of route directions through navigational performance. *Spatial Cogn. & Compn.* , 3(4):269–289.
- Davis, J. R. (1986). Giving directions: A voice interface to an urban navigation program. *American Voice I/O Society*, pages 77–84.
- Dayan, A. and Thomas, J. R. (1995). Development of automatic and effortful processes in memory for spatial location of movement. *Human Performance*, 8(1):51–66.
- Denis, M. (1997). The description of routes: A cognitive approach to the production of spatial discourse. *Current Psych. of Cogn.* , 16(4):409–458.

- Denis, M., Pazzaglia, F., Cornoldi, C., and Bertolo, L. (1999). Spatial discourse and navigation: An analysis of route directions in the city of Venice. *Applied Cog. Psych.*, 13(2):145–174.
- Di Eugenio, B. (1992). Understanding natural language instructions: the case of purpose clauses. In *Proc. of 30th Ann. Meeting of the ACL (ACL-92)*, pages 120–127, Newark, DE. Morgan Kaufmann.
- Di Eugenio, B. (1998). An action representation formalism to interpret natural language instructions. *Compl. Intelligence*, 14(1):89–133.
- Dixon, P. (1987a). The processing of organizational and component step information in written directions. *J. of Memory & Lang.*, 26(1):24–35.
- Dixon, P. (1987b). The structure of mental plans for following directions. *J. of Exptl. Psych.*, 13(1):18–26.
- Dixon, P., Faries, J., and Gabrys, G. (1988). The role of explicit action statements in understanding and using written directions. *J. of Memory & Lang.*, 27(6):649–667.
- Duckham, M. and Kulik, L. (2003). “Simplest” paths: Automated route selection for navigation. In Kuhn, W., Worboys, M. F., and Timpf, S., editors, *Proc. of COSIT-03*, number 2825 in LNCS, pages 169–185, Kartause Ittingen, Switzerland. Springer-Verlag.
- Edmonds, P. G. (1993). A computational model of collaboration on reference in direction-giving dialogues. Master’s thesis, Univ. of Toronto, Dept. of Comp. Sci., Toronto, Canada. Published as Technical Report CSRI-289.
- Edmonds, P. G. (1994). Collaboration on reference to objects that are not mutually known. In *Proc. of 15th Intl. Conf. on Compl. Ling. (COLING-94)*, pages 1118–1122, Kyoto, Japan.

- Elliot, R. J. and Lesk, M. E. (1982). Route finding in street maps by computers and people. In *Proc. of 2nd Natl. Conf. on AI (AAAI-82)*, pages 258–261, Pittsburgh, PA.
- Ellsworth, M., Erk, K., Kingsbury, P., and Pado, S. (2004). PropBank, SALSA, and FrameNet: How design determines product. In *Proc. of LREC 2004 Ws. on Building Lexical Resources from Semantically Annotated Corpora*, pages 17–23, Lisbon, Portugal.
- Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- Ferguson, G. and Allen, J. F. (1993). Generic plan recognition for dialogue systems. In *Proc. of Ws. on Human Language Technology*, pages 171–176, Plainsboro, New Jersey.
- Ferguson, G. and Allen, J. F. (1998). TRIPS: An integrated intelligent problem-solving assistant. In *Proc. of 15th Natl. Conf. on AI (AAAI-98)*, Madison, WI.
- Firby, R. J. (1989). *Adaptive Execution in Complex Dynamic Worlds*. PhD thesis, Yale Univ., Comp. Sci. Dept., New Haven, CT.
- Fischer, K. (2003). Linguistic methods for investigating concepts in use. In Stolz, T. and Kolbe, K., editors, *Methodologie in der Linguistik*, pages 39–62. Peter Lang, Frankfurt a.M., Germany.
- Fleischman, M. and Hovy, E. (2006). Taking advantage of the situation: Non-linguistic context for natural language interfaces to interactive virtual environments. In *Proc. of 11th Intl. Conf. on Intelligent User Interfaces (IUI '06)*, Sydney, Australia.
- Fleischman, M. and Roy, D. (2005). Intentional context in situated natural language learning. In *Proc. of Conf. on Natural Language Learning*, Ann Arbor, MI.
- Fong, T. and Nourbakhsh, I. (2005). Interaction challenges in human-robot space exploration. *ACM Interactions*.

- Fontaine, S. and Denis, M. (1999). The production of route instructions in underground and urban environments. In COSIT-99 (1999), pages 83–94.
- Fox, D., Burgard, W., and Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. *J. of AI Res.* , 11:391–427.
- Fraczak, L., Lapalme, G., and Zock, M. (1998). Automatic generation of subway directions: Saliency gradation as a factor for determining message and form. In Hovy, E., editor, *Proc. of Ninth Intl. Ws. on Nat. Lang. Gen.* , pages 58–67. Assoc. Compl. Lang. , New Brunswick, New Jersey.
- Frank, A. U. (2003). Pragmatic information content: How to measure the information in a route description. In Duckham, M., editor, *Foundations of Geog. Info. Sci.* , pages 47–68. Taylor & Francis.
- Freksa, C., Brauer, W., Habel, C., and Wender, K. F., editors (2000). *Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, volume 1849 of *LNCS*. Springer.
- Freksa, C., Brauer, W., Habel, C., and Wender, K. F., editors (2003). *Spatial Cognition III: Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning*, volume 2685 of *LNCS*. Springer.
- Freksa, C., Habel, C., and Wender, K. F., editors (1998). *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, volume 1404 of *LNCS*, Berlin. Springer.
- Freksa, C., Knauff, M., Krieg-Brückner, B., Nebel, B., and Barkowsky, T., editors (2004). *Spatial Cognition IV: Reasoning, Action, Interaction: International Conference Spatial Cognition 2004*, volume 3343 of *LNCS*, Frauenchiemsee, Germany. Springer; Berlin.
- Freundschuh, S. and Egenhofer, M. (1997). Human conceptions of spaces: Implications for GIS. *Trans. on Geog. Info. Sci.* , 2(4):361–375.

- Garden, S., Cornoldi, C., and Logie, R. H. (2002). Visuo-spatial working memory in navigation. *Applied Cog. Psych.*, 16(1):35–50.
- Garrod, S. (1989). Conceptual and semantic co-ordination in dialogue: Implications for the design of interactive natural language interfaces. In Peckham, J., editor, *Recent Developments and Applications of Natural Language Processing*, UNICOM applied information technology reports, pages 262–272. Kogan Page.
- Garrod, S. and Anderson, A. (1987). Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, 27(2):181–218.
- Garrod, S. and Doherty, G. (1994). Conversation, co-ordination and convention: An empirical investigation of how groups establish linguistic conventions. *Cognition*, 53:181–215.
- Garrod, S. C. and Sanford, A. J. (1989). Discourse models as interfaces between language and the spatial world. *J. of Semantics*, 6:147–160.
- Ge, R. and Mooney, R. J. (2005). A statistical semantic parser that integrates syntax and semantics. In *Proc. of the Ninth Conf. on Computational Natural Language Learning*, pages 9–16, Ann Arbor, MI.
- Geldof, S. (2003). Corpus analysis for NLG. In Reiter, E., Horacek, H., and van Deemter, K., editors, *Proc. of 9th European Ws. on Nat. Lang. Gen. Comp. Sci. Conf. on (ENLG'03)*, Budapest, Hungary.
- Gildea, D. and Hockenmaier, J. (2003). Identifying semantic roles using combinatory categorial grammar. In *2003 Conf. on Empirical Methods in Nat. Lang. Proc. (EMNLP)*, pages 57–64, Sapporo, Japan.
- Gildea, D. and Palmer, M. (2002). The necessity of syntactic parsing for predicate argument recognition. In *Proc. of 40th Ann. Meeting of the ACL (ACL-02)*, pages 239–246, Philadelphia, PA.

- Goan, T., Benson, N., and Etzioni, O. (1996). A grammar inference algorithm for the world wide web. In *Proc. of AAAI Spring Symp. on Machine Learning in Information Access*, Stanford, CA.
- Golding, J. M., Graesser, A. C., and Hauselt, J. (1996). The process of answering direction-giving questions when someone is lost on a university campus: The role of pragmatics. *Applied Cog. Psych.*, 10(1):23–39.
- Goodrich, M. A., Schultz, A. C., and Bruemmer, D. J., editors (2006). *Proc. of 1st ACM Conf. on Human-Robot Interaction*, Salt Lake City, UT.
- Gorniak, P. and Roy, D. (2004). Grounded semantic composition for visual scenes. *J. of AI Res.*, 21:429–470.
- Gorniak, P. and Roy, D. (2006). Perceived affordances as a substrate for linguistic concepts. In *Proc. of 28th Ann. Meeting of the Cog. Sci. Society (CogSci-06)*, Vancouver, BC.
- Gorniak, P. and Roy, D. (2007). Situated language understanding as filtering perceived affordances. *Cog. Sci.*. In Press.
- Green, A., Hüttenrauch, H., Topp, E. A., and Eklundh, K. S. (2006). Developing a contextualized multimodal corpus for human-robot interaction. In *Proc. of 5th Intl. Conf. on Language Resources and Evaluation (LREC2006)*, Genoa, Italy.
- Grice, H. P. (1967). Logic and conversation. William James Lectures, Harvard Univ.. Published in Grice (1989).
- Grice, H. P. (1975). Logic and conversation. In Cole, P. and Morgan, J. L., editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 43–58. Academic Press, New York.
- Grice, H. P. (1989). *Studies in the Way of Words*. Harvard Univ. Press, Cambridge, MA.

- Grodner, D. J. and Sedivy, J. C. (2004). The effect of speaker-specific information on pragmatic inferences. In Gibson, N. P. . E., editor, *The Processing and Acquisition of Reference*. MIT Press.
- Gruenstein, A. (2002). Conversational interfaces: A domain-independent architecture for task-oriented dialogues. Master's thesis, Stanford Univ., Symbolic Sys. Program, Stanford, CA.
- Gryl, A., Moulin, B., and Kettani, D. (2002). A conceptual model for representing verbal expressions used in route directions. In Coventry and Oliver (2002), pages 19–42.
- Guhe, M., Habel, C., and Tschander, L. (2003). Describing motion events: Incremental representations for incremental processing. In *Proc. of 5th Intl. Ws. on Compl. Semantics (IWCS-5)*, pages 410–424, Tilburg, The Netherlands.
- Gupta, R. and Hennacy, K. (2005). Commonsense reasoning about task instructions. In Thórisson, K. R., Vilhjalmsón, H., and Marsella, S. C., editors, *Proc. of AAAI Ws. on Modular Construction of Human-Like Intelligence*, pages 86–91, Pittsburgh, PA. AAAI Press.
- Gupta, R. and Kochenderfer, M. J. (2004). Common sense data acquisition for indoor mobile robots. In *Proc. of 19th Natl. Conf. on AI (AAAI-2004)*, Menlo Park, CA.
- Habel, C. (2003). Incremental generation of multimodal route instructions. In *AAAI Spring Symp. on Natural language generation in spoken and written dialogue*, Stanford, CA.
- Haigh, K. Z., Shewchuk, J. R., and Veloso, M. M. (1997). Exploiting domain geometry in analogical route planning. *J. of Exptl. and Theo. AI*, 9:509–541.
- Hansen, S., Richter, K.-F., and Klippel, A. (2006). Landmarks in OpenLS: A data structure for cognitive ergonomic route directions. In Raubal, M., Miller, H., Frank, A. U., and Goodchild, M. F., editors, *Proc. of GIScience*, number 4197 in LNCS, Berlin.

- Hayward, W. G. and Tarr, M. J. (1995). Spatial language and spatial representation. *Cognition*, 55(1):39–84.
- Heeman, P. A. and Hirst, G. (1992). Collaborating on referring expressions. Technical Report CSRI-289, Comp. Sci. Dept. Univ. of Rochester, Rochester, NY.
- Herskovits, A. (1985). Semantics and pragmatics of locative expressions. *Cog. Sci.*, 9(3):341–378.
- Herskovits, A. (1997). Language, spatial cognition, and vision. In Stock, O., editor, *Spatial and Temporal Reasoning*, pages 155–202. Kluwer Academic Publishers, Boston, MA.
- Huffman, S. B. and Laird, J. E. (1995). Flexibly instructable agents. *J. of AI Res.*, 3:271–324.
- Hüttenrauch, H., Green, A., Norman, M., Oestreicher, L., and Eklundh, K. S. (2004). Involving users in the design of a mobile office robot. *IEEE Trans. on Sys., Man & Cybernetics – Part C: App. & Reviews*, 34(2):113–124.
- Jackendoff, R. (1983). *Semantics and Cognition*. MIT Press.
- Jackson, P. G. (1998). In search of better route instructions. *Ergonomics*, 41(7):1001–1013.
- Johnson, C. and Fillmore, C. J. (2000). The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proc. of North American Assoc. Compl. Lang.*
- Jung, H., Allen, J., Chambers, N., Galescu, L., Swift, M., and Taysom, W. (2006). One-shot procedure learning from instruction and observation. In *Proc. of Intl. FLAIRS Conf.: Special Track on Natl. Lang. & Knowledge Representation*.
- Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *AI*, 101:99–134.

- Kamp, H. and Reyle, U. (1993). *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. Kluwer Academic Publishers, London, Boston, Dordrecht. Studies in Linguistics and Philosophy, Volume 42.
- Kate, R. J., Wong, Y. W., and Mooney, R. J. (2005). Learning to transform natural to formal languages. In *Proc. of 20th Natl. Conf. on AI (AAAI-2005)*, pages 1062–1068, Pittsburgh, PA.
- Kelleher, J., Costello, F. J., and van Genabith, J. (2005). Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context. *AI*, 167(1–2):62–102. Connecting Language to the World.
- Kingsbury, P., Palmer, M., and Marcus, M. (2002). Adding semantic annotation to the Penn Treebank. In *Proc. of Human Lang. Technology*, San Diego, CA.
- Klippel, A., Tappe, H., and Habel, C. (2003). Pictorial representations of routes: Chunking route segments during comprehension. In Freksa et al. (2003), pages 11–33.
- Klippel, A., Tappe, H., Kulik, L., and Lee, P. U. (2005). Wayfinding choremes - a language for modeling conceptual route knowledge. *J. of Visual Languages & Computing*, 16(4):311–329.
- Klippel, A. and Winter, S. (2005). Structural salience of landmarks for route directions. In Cohn, A. G. and Mark, D. M., editors, *Proc. of COSIT-05*, number 3693 in LNCS, pages 347–362, Ellicottville, New York. Springer-Verlag.
- Knees, M. H. (2002). Designing an anaphora resolution algorithm for route instructions. Master's thesis, Division of Informatics, Univ. of Edinburgh. M.Sc. in Cognitive Science and Natural Language.
- Koenig, S. and Simmons, R. G. (1996). Unsupervised learning of probabilistic models for

- robot navigation. In *Proc. of IEEE Intl. Conf. on Rob. & Autom. (ICRA-96)*, pages 2301–2308, Los Alamitos, CA. IEEE Computer Society Press.
- Koenig, S. and Simmons, R. G. (1998). Xavier: a robot navigation architecture based on partially observable Markov decision process models. In Kortenkamp, D., Bonasso, R. P., and Murphy, R., editors, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, pages 91–122. AAAI Press/The MIT Press, Menlo Park, CA.
- Kopp, S., Tepper, P. A., Ferriman, K., and Cassell, J. (2007). Trading spaces: How humans and humanoids use speech and gesture to give directions. *Spatial Cogn. & Compn.* . in press.
- Kortenkamp, D., MacMahon, M., Ryan, D., Bonasso, R. P., and Moreland, L. (1998). Applying a layered control architecture to a free-flying space camera. In *Proc. of IEEE Intl. Joint Symposia on Intelligence & Sys.* , pages 188–194, Rockville, MD, USA.
- Krieg-Brückner, B., Frese, U., Lüttich, K., Mandel, C., Mossakowski, T., and Ross, R. J. (2004). Specification of an ontology for route graphs. In Freksa et al. (2004), pages 390–412.
- Krieg-Brückner, B., Röfer, T., Carmesin, H.-O., and Müller, R. (1998). A taxonomy of spatial knowledge for navigation and its application to the Bremen autonomous wheelchair. In Freksa et al. (1998), pages 373–398.
- Kuipers, B. and Kassirer, J. (1987). Knowledge acquisition by analysis of verbatim protocols. In Kidd, A., editor, *Knowledge Acquisition for Expert Systems*. Plenum, New York.
- Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., and Savelli, F. (2004). Local metrical and global topological maps in the Hybrid Spatial Semantic Hierarchy. In *Proc. of IEEE Intl. Conf. on Rob. & Autom. (ICRA-04)*, New Orleans, LA. IEEE Computer Society Press.

- Kuipers, B. J. (2000). The Spatial Semantic Hierarchy. *AI*, 119:191–233.
- Kuipers, B. J. (2006). An intellectual history of the Spatial Semantic Hierarchy. In Jefferies, M. and Yeap, A. W.-K., editors, *Robot and Cognitive Approaches to Spatial Mapping*. Springer-Verlag.
- Kuipers, B. J. and Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *J. of Rob. & Auton. Sys.*, 8:47–63.
- Kuipers, B. J., Tecuci, D. G., and Stankiewicz, B. J. (2003). The skeleton in the cognitive map: A computational and empirical exploration. *Env. & Behavior*, 35(1):80–106.
- Kyriacou, T., Bugmann, G., and Lauria, S. (2002). Vision-based urban navigation procedures for verbally instructed robots. In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots & Sys. (IROS-02)*.
- Kyriacou, T., Bugmann, G., and Lauria, S. (2004). Vision-based urban navigation procedures for verbally instructed robots. *Rob. & Auton. Sys.*. XXX.
- Landau, B. and Jackendoff, R. (1993). ‘What’ and ‘Where’ in spatial language and spatial cognition. *The Behavioral & Brain Scis.*, 16(2):217–265.
- Lankenau, A. and Röfer, T. (2001). A versatile and safe mobility assistant. *IEEE Robotics & Autom. Magazine*, 8(1):29–37.
- Lauria, S., Bugmann, G., Kyriacou, T., Bos, J., and Klein, E. (2001). Training personal robots using natural language instruction. *IEEE Intelligent Sys.*, pages 2–9. XXX.
- Lauria, S., Bugmann, G., Kyriacou, T., and Klein, E. (2002a). Mobile robot programming using natural language. *Rob. & Auton. Sys.*, 38(3–4):171–181.
- Lauria, S., Kyriacou, T., Bugmann, G., Bos, J., and Klein, E. (2002b). Converting natural language route instructions into robot-executable procedures. In *Proc. of IEEE*

- Intl. Ws. on Robot and Human Interactive Communication (RO-MAN)*, pages 223–228, Berlin, Germany.
- Lawton, C. A. (2001). Gender and regional differences in spatial referents used in direction giving. *Sex Roles*, 44(5):321–337.
- Lemon, O., Gruenstein, A., and Peters, S. (2002). Collaborative activities and multi-tasking in dialogue systems: Towards natural dialogue with robots. *Traitement Automatique des Langues (TAL)*, 43(2):131–154. Spec. Issue on Dialogue.
- Levelt, W. J. M. (1982). Cognitive styles in the use of spatial direction terms. In Jarvalla, R. J. and Klein, W., editors, *Speech, Place, and Action: Studies in Deixis and Related Topics*, pages 251–268. Wiley, Chichester.
- Levit, M. and Roy, D. (2007). Interpretation of spatial language in a map navigation task. *IEEE Trans. on Sys., Man & Cybernetics – Part B: Cybernetics*. in press.
- Li, H. and Abe, N. (1998). Generalizing case frames using a thesaurus and the MDL principle. *Compl.*, 24(2):217–244.
- Lieberman, H. and Mausby, D. (1996). Instructible agents: Software that just keeps getting better. *IBM Systems Journal*, 35(3–4):539–556.
- Linde, C. (1974). *The Linguistic Encoding of Spatial Information*. PhD thesis, Columbia University.
- Linde, C. and Labov, W. (1975). Spatial structures as a site for the study of language and thought. *Language*, 51(4):924–939.
- Liu, H. and Lieberman, H. (2005). Programmatic semantics for natural language interfaces. In *Proc. of ACM Conf. on Human Factors in Computing Systems (CHI '05)*, pages 1597–1600, Portland, OR.

- Lizogat, G. (2000). From language to motion, and back: Generating and using route descriptions. In Christodoulakis, D., editor, *Natural Language Processing*, volume 1835 of *LNCS*, pages 328–345. Springer, Patras, Greece.
- Lovelace, K. L., Hegarty, M., and Montello, D. R. (1999). Elements of good route directions in familiar and unfamiliar environments. In *COSIT-99 (1999)*, pages 56–82.
- Lundh, F. (1999). An introduction to Tkinter. <http://www.pythonware.com/library/tkinter/introduction/>.
- Maass, W. (1994). From vision to multimodal communication: Incremental route descriptions. *AIRvw.*, 8(2-3):159–174.
- Maass, W. (1995). How spatial information connects visual perception and natural language generation in dynamic environments. In Frank, A. U. and Kuhn, W., editors, *Proc. of COSIT-95*, volume 988 of *LNCS*, pages 223–240, Semmering, Austria. Springer.
- MacMahon, M. (2005). Understanding and following route instructions through large-scale space. In *Proc. of Ws. on Spatial Language and Dialogue*, Delmenhorst, Germany.
- MacMahon, M., Adams, W., Bugajska, M., Perzanowski, D., Schultz, A., and Thomas, S. (2004). Adjustable autonomy for route-direction following. In *Proc. of AAAI Spring Symp. on Interaction between Humans & Auton. Systems over Extended Operation*, Stanford, CA.
- MacMahon, M. and Stankiewicz, B. (2006). Human and automated indoor route instruction following. In *Proc. of 28th Ann. Meeting of the Cog. Sci. Society (CogSci-06)*, pages 1759–1764, Vancouver, BC.
- MacMahon, M., Stankiewicz, B., and Kuipers, B. (2006). Walk the talk: Connecting language, knowledge, action in route instructions. In *Proc. of 21st Natl. Conf. on AI (AAAI-2006)*, pages 1475–1482, Boston, MA.

- Mandel, C., Huebner, K., and Vierhuff, T. (2005). Towards an autonomous wheelchair: Cognitive aspects in service robotics. In *Proc. of Towards Autonomous Robotic Systems (TAROS 2005)*, pages 165–172, London, UK.
- Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.
- McClosky, D., Charniak, E., and Johnson, M. (2006). Effective self-training for parsing. In *Proc. of North American Assoc. Compl. Lang.* .
- McDermott, D. and Davis, E. (1984). Planning routes through uncertain territory. *AI*, 22:107–156.
- Mellet, E., Briscogne, S., Tzourio-Mazoyer, N., Ghaem, O., Petit, L., Zago, L., Etard, O., Berthoz, A., Mazoyer, B., and Denis, M. (2000). Neural correlates of topographic mental exploration: The impact of route versus survey perspective learning. *NeuroImage*, 12(5):588–600.
- Michon, P.-E. and Denis, M. (2001). When and why are visual landmarks used in giving directions? In Montello (2001), pages 292–305.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Boston.
- Modayil, J., Beeson, P., and Kuipers, B. (2004). Using the topological skeleton for scalable global metrical map-building. In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots & Sys. (IROS-04)*, pages 1530–1536, Sendai, Japan.
- Modayil, J. and Kuipers, B. (2006). Autonomous shape model learning for object localization and recognition. In *Proc. of IEEE Intl. Conf. on Rob. & Autom.* , pages 2991–2996.
- Montello, D. R., editor (2001). *Spatial Information Theory: Foundations of Geog. Info. Sci. (COSIT '01)*, volume 2205 of *LNCS*, Morro Bay, CA. Springer.

- Moratz, R. and Tenbrink, T. (2006). Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations. *Spatial Cogn. & Compn.* , 6(1):63–106.
- Moratz, R., Tenbrink, T., Bateman, J., and Fischer, K. (2003). Spatial knowledge representation for human-robot interaction. In Freksa et al. (2003), pages 263–286.
- Moravec, H. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proc. of IEEE Intl. Conf. on Rob. & Autom.* , pages 116–121.
- Moulin, B. and Kettani, D. (1998). Combining a logical and an analogical framework for route generation and description. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):155–179.
- Mukerjee, A. (1998). Neat vs scruffy: A survey of computational models for spatial expressions. In Olivier and Gapp (1998).
- Müller, R., Röfer, T., Lankenau, A., Musto, A., Stein, K., and Eisenkolb, A. (2000). Coarse qualitative descriptions in robot navigation. In Freksa et al. (2000), pages 265–276.
- Murphy, R. R. (2004). Human-robot interaction in rescue robotics. *IEEE Trans. on Sys. , Man & Cybernetics – Part C:App. & Reviews*, 34(2):138–153.
- Narayanan, S. (1997). Talking the talk is like walking the walk: A computational model of verbal aspect. In *Proc. of 19th Ann. Meeting of the Cog. Sci. Society (CogSci-97)*, Stanford, CA.
- Nicolescu, M. and Mataric, M. J. (2003). Natural methods for learning and generalization in human-robot domains. In *Proc. of the Second Intl. Conf. on Auton. Agents*, Melbourne, Australia.
- Nothegger, C., Winter, S., and Raubal, M. (2004). Selection of salient features for route directions. *Spatial Cogn. & Compn.* , 4(2):113–136.

- Olivier, P. and Gapp, K.-P., editors (1998). *Representation and processing of spatial expressions*. Lawrence Erlbaum Associates, Mahwah, NJ.
- Palmer, M., Kingsbury, P., and Gildea, D. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Compl. Ling.* , 31(1):711–106.
- Perrault, C. R. and Allen, J. F. (1980). A plan-based analysis of indirect speech acts. *American J. of Compl. Ling.* , 6(3-4).
- Perzanowski, D., Brock, D., Adams, W., Bugajska, M., Schultz, A. C., Trafton, J. G., Blisard, S., and Skubic, M. (2003). Finding the FOO: A pilot study for a multimodal interface. In *Proc. of IEEE Sys., Man & Cybernetics Conf.*, pages 3218–3223, Washington, DC.
- Perzanowski, D., Schultz, A., Adams, W., Bugajska, M., Abramson, M., MacMahon, M., Atrash, A., and Coblenz, M. (2002). “Excuse me, where’s the registration desk?”: Report on integrating systems for the Robot Challenge AAAI 2002. In *Human-Robot Interaction, Papers from the 2002 AAAI Fall Symp.*, pages 63–72. AAAI Press. Technical Report FS-02-03.
- Perzanowski, D., Schultz, A. C., and Adams, W. (1998). Integrating natural language and gesture in a robotics domain. In *Proc. of Intl. Symp. on Intelligent Control*, pages 247–252, Washington, DC. IEEE Computer Society Press.
- Perzanowski, D., Schultz, A. C., Adams, W., Marsh, E., and Bugajska, M. (2001). Building a multimodal human-robot interface. *IEEE Intelligent Sys.* , pages 16–21.
- Pickering, M. J. and Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *The Behavioral & Brain Scis.* , 27(2):169226.
- Pook, P. K. and Ballard, D. H. (1996). Deictic human/robot interaction. *Rob. & Auton. Sys.* , 18(1–2):259–269. Proc. of Intl. Ws. on Biorobotics: Human-Robot Symbiosis.

- Porzel, R., Jansche, M., and Meyer-Klabunde, R. (2002). Generating spatial descriptions from a cognitive point of view. In Coventry and Oliver (2002), pages 185–208.
- Purver, M., Ratiu, F., and Cavedon, L. (2006). Robust interpretation in dialogue by combining confidence scores with contextual features. In *Proc. of Intl. Conf. on Spoken Language Processing (Interspeech/ICSLP)*, pages 1–4, Pittsburgh, PA.
- Python (2007). Python programming language.
- Raubal, M. and Winter, S. (2002). Enriching wayfinding instructions with local landmarks. In Egenhofer, M. J. and Mark, D. M., editors, *Proc. of GIScience 2002*, volume 2478 of *LNCS*, pages 243–259, Boulder, CO. Springer.
- Regier, T. and Carlson, L. A. (2001). Grounding spatial language in perception: An empirical and computational investigation. *J. of Exptl. Psych.* , 130(2):273–298.
- Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *J. of Natural Lang. Eng.* , 1(1):1–32.
- Remolina, E. and Kuipers, B. (2004). Towards a general theory of topological maps. *AI* , 152(1):47–104.
- Richter, K.-F., Klippel, A., and Freksa, C. (2004). Shortest, fastest, - but what next? a different approach to route directions. In Raubal, M., Sliwinski, A., and Kuhn, W., editors, *Geoinformation und Mobilität - von der Forschung zur praktischen Anwendung. Beiträge zu den Münsteraner GI-Tagen 2004*, IfGIprints, pages 205–217. Institut für Geoinformatik; Münster.
- Riesbeck, C. (1980). “You can’t miss it!”: Judging the clarity of directions. *Cog. Sci.* , 4:285–303.
- Rogers, S., Fiechter, C.-N., and Langley, P. (1999). An adaptive interactive agent for route advice. In *Proc. of Third Intl. Conf. on Auton. Agents*, pages 198–205. ACM Press.

- Rosenthal, R. and Rosnow, R. L. (1991). *Essentials of Behavioral Research: Methods and Data Analysis*. McGraw-Hill, 2 edition.
- Ross, R. J., Shi, H., Vierhuff, T., Krieg-Brückner, B., and Bateman, J. (2004). Towards dialogue based shared control of navigating robots. In Freksa et al. (2004), pages 478–499.
- Roy, D. (2005). Semiotic schemas: a framework for grounding language in action and perception. *AI*, 167(1–2):170–205.
- Roy, D., Hsiao, K.-Y., and Mavridis, N. (2004). Mental imagery for a conversational robot. *IEEE Trans. on Sys., Man & Cybernetics – Part B: Cybernetics*, 34(3):1374–1383.
- Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, 1st edition.
- Sandstrom, N. J., Kaufman, J., and Huettel, S. A. (1998). Males and females use different distal cues in a virtual environment navigation task. *Cog. Brain Research*, 6(4):351–360.
- Scherl, R. B. and Levesque, H. J. (2003). Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1-2):1–39.
- Schmitz, S. (1999). Gender differences in acquisition of environmental knowledge related to wayfinding behavior, spatial anxiety and self-estimated environmental competencies. *Sex Roles*, 41(1-2):71–93.
- Schober, M. F. (1993). Spatial perspective-taking in conversation. *Cognition*, 47(1):1–24.
- Sedivy, J. C. (2003). Pragmatic versus form-based accounts of referential contrast: Evidence for effects of informativity expectations. *J. of Psycholinguistic Research*, 32(1):3–23.
- Shi, H. and Tenbrink, T. (2005). Telling Rolland where to go: HRI dialogues on route navigation. In *Proc. of WoSLaD Workshop on Spatial Language and Dialogue*.

- Shimizu, N. and Haas, A. (2006). Extracting frame-based knowledge representation from route instructions. In *HLT-NAACLWs. on Computationally Hard Problems and Joint Inference in Speech and Language Processing*. Late Breaking Paper.
- Siegel, A. W. and White, S. H. (1975). The development of spatial representations of large-scale environments. In Reese, H. W., editor, *Advances in Child Development and Behavior*, volume 10, pages 9–55. Academic Press, New York.
- Simmons, R. and Apfelbaum, D. (1998). A task description language for robot control. In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots & Sys. (IROS-98)*.
- Simmons, R., Goldberg, D., Goode, A., Montemerlo, M., Roy, N., Sellner, B., Urmson, C., Schultz, A., Abramson, M., Adams, W., Atrash, A., Bugajska, M., Coblenz, M., MacMahon, M., Perzanowski, D., Horswill, I., Zubek, R., Kortenkamp, D., Wolfe, B., Milam, T., and Maxwell, B. (2003). GRACE: An autonomous robot for the AAI Robot Challenge. *AI Magazine*, 24(2):51–72.
- Simmons, R. G. and Koenig, S. (1995). Probabilistic robot navigation in partially observable environments. In *Proc. of 14th Intl. Joint Conf. on AI (IJCAI-95)*, pages 1080–1087, Montreal, Canada. Intl. Joint Conf. on AI, AAAI Press/The MIT Press.
- Simpson, R. C. (2005). Smart wheelchairs: A literature review. *J. of Rehabilitation Research & Development*, 42(4):423–436.
- Siskind, J. M. (1990). Acquiring core meanings of words, represented as jackendoff-style conceptual structures, from correlated streams of linguistic and non-linguistic input. In *Proc. of 28th Ann. Meeting of the ACL (ACL-90)*, pages 143–156, Pittsburgh, PA.
- Siskind, J. M. (1995). Grounding language in perception. *AI Rvw.* , 8(5-6):371–391.
- Siskind, J. M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *J. of AI Res.* , 15:31–90.

- Skantze, G. (2005). Exploring human error recovery strategies: Implications for spoken dialogue systems. *Speech Communication*, 45(3):325–341. Spec. Issue on Error Handling in Spoken Dialogue Systems.
- Skubic, M., Blisard, S., Bailey, C., Adams, J., and Matsakis, P. (2004a). Qualitative analysis of sketched route maps: Translating a sketch into linguistic descriptions. *IEEE Trans. on Sys., Man & Cybernetics – Part B: Cybernetics*, 34(2):1275–1282.
- Skubic, M., Matsakis, P., Forrester, B., and Chronis, G. (2001). Generating linguistic spatial descriptions from sonar readings using the histogram of forces. In *Proc. of IEEE Intl. Conf. on Rob. & Autom. (ICRA-01)*.
- Skubic, M., Perzanowski, D., Blisard, S., Schultz, A., Adams, W., Bugajska, M., and Brock, D. (2004b). Spatial language for human-robot dialogs. *IEEE Trans. on Sys., Man & Cybernetics – Part C: App. & Reviews*, 34(2):154–167.
- Sorrows, M. E. and Hirtle, S. C. (1999). The nature of landmarks for real and electronic spaces. In *COSIT-99 (1999)*, pages 37–50.
- Sperber, D. and Wilson, D. (1986). *Relevance: Communication and Cognition*. Blackwell Publishers, Oxford.
- Sperber, D. and Wilson, D. (2004). Relevance Theory. In Horn, L. R. and Ward, G., editors, *The Handbook of Pragmatics*, pages 607–632. Blackwell, Oxford.
- Stankiewicz, B. and Eastman, K. (2008). Lost in Virtual Space II: The role of proprioception and discrete actions when navigating with uncertainty. *ACM Trans. on Applied Perception*. under review.
- Stankiewicz, B. and Kalia, A. (2007). Acquisition and retention of structural versus object landmark knowledge when navigating through a large-scale space. *J. of Exptl. Psych. : Human Perception & Performance*. in press.

- Stankiewicz, B. J., Legge, G. E., Mansfield, J. S., and Schlicht, E. J. (2006). Lost in virtual space: Studies in human and ideal spatial navigation. *J. of Exptl. Psych. : Human Perception & Performance*, 32(3):686–704.
- Stankiewicz, B. J., Legge, G. E., and Schlicht, E. (2001). The effect of layout complexity on human and ideal navigation performance. *J. of Vision*, 1(3).
- Stocky, T. A. (2002). Conveying routes: Multimodal generation and spatial intelligence in embodied conversational agents. Master's thesis, Mass. Inst. of Technology, Cambridge, MA.
- Stoia, L., Byron, D., Shockley, D., and Fosler-Lussier, E. (2006). Sentence planning for realtime navigational instruction. In *Proc. of Human Language Technology Conf. of the NAACL*, pages 157–160, New York City, USA.
- Stone, M., Doran, C., Webber, B., Bleam, T., and Palmer, M. (2003). Microplanning with communicative intentions: The SPUD system. *Compl. Intelligence*, 19(4):311–381.
- Striegnitz, K., Tepper, P., Lovett, A., and Cassell, J. (2005). Knowledge representation for generating locating gestures in route directions. In *Proc. of WoSLaD Workshop on Spatial Language and Dialogue*, Delmenhorst, Germany.
- Talmy, L. (1983). How language structures space. In H. L. Pick, J. and Acredolo, L. P., editors, *Spatial orientation : Theory, research and application*, pages 225–282. Plenum, NY.
- Talmy, L. (2000). *Toward A Cognitive Semantics*, volume I of *Language, Speech, and Communication*. MIT Press, Cambridge, MA.
- Tappe, H. and Habel, C. (1998). Verbalization of dynamic sketch maps: Layers of representation and their interaction. In *Proc. of 20th Ann. Meeting of the Cog. Sci. Society (CogSci-98)*, Madison, WI.

- Taylor, H. A. and Tversky, B. (1992). Spatial mental models derived from survey and route descriptions. *J. of Memory & Lang.*, 31(2):261–292.
- Taylor, H. A. and Tversky, B. (1996). Perspective in spatial descriptions. *J. of Memory & Lang.*, 35(3):371–391.
- Taylor, H. A., Uttal, D. H., Fisher, J., and Mazepa, M. (2001). Ambiguity in acquiring spatial representation from descriptions compared to depictions: The role of spatial orientation. In Montello (2001), pages 278–291.
- Tellex, S. and Roy, D. (2006). Spatial routines for a simulated speech-controlled vehicle. In Goodrich et al. (2006).
- Tellex, S. and Roy, D. (2007). Grounding language in spatial routines. In *Proc. of AAAI Spring Symp. on on Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems*, Stanford, CA.
- Tenbrink, T. (2003). Conveying spatial information in linguistic human-robot interaction. In *Proc. of Ws. on Semantics and Pragmatics of Dialogue*, pages 207–8.
- Tenbrink, T., Fischer, K., and Moratz, R. (2002). Spatial strategies in linguistic human-robot communication. In Freksa, C., editor, *Künstliche Intelligenz Themenheft 4/02 Spatial Cognition*, pages 19–23. arenDTaP Verlag.
- Tenbrink, T. and Klippel, A. (2005). Achieving reference via contrast in route instructions and spatial object identification. In *Ws. on Reference, 21st Scandinavian Conference of Linguistics*.
- Tenbrink, T. and Moratz, R. (2003). Group-based spatial reference in linguistic human-robot interaction. In *Proc. of European Cognitive Science Conf. on (EuroCogSci-03)*, Osnabrück, Germany.

- Tews, A. D., Matarić, M. J., and Sukhatme, G. S. (2003). A scalable approach to human-robot interaction. In *Proc. of IEEE Intl. Conf. on Rob. & Autom.*, pages 1665–1670, Taipei, Taiwan.
- Theocharous, G., Murphy, K., and Kaelbling, L. P. (2004). Representing hierarchical POMDPs as DBNs for multi-scale robot localization. In *Proc. of IEEE Intl. Conf. on Rob. & Autom. 2004*.
- Timpf, S., Volta, G. S., Pollock, D. W., and Egenhofer, M. J. (1992). A conceptual model of wayfinding using multiple levels of abstraction. In Campari, I., Frank, A. U., and Formentini, U., editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, volume 639, pages 348–367. Springer-Verlag, Lecture Annotate in Computer Science.
- Trafton, J. G., Cassimatis, N. L., Bugajska, M. D., Brock, D. P., Mintz, F. E., and Schultz, A. C. (2005). Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Trans. on Sys., Man & Cybernetics – Part A: Sys. & Humans*, 35(4):460–470. Spec. Issue on Human-Robot Interaction.
- Tschander, L. B., Schmidtke, H. R., Eschenbach, C., Habel, C., and Kulik, L. (2003). A geometric agent following route instructions. In Freksa et al. (2003), pages 89–111.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.
- Tversky, B. (2000). Some ways that maps and diagrams communicate. In Freksa et al. (2000), pages 72–79.
- Tversky, B., Lee, P., and Mainwaring, S. (1999). Why do speakers mix perspectives? *Spatial Cogn. & Compn.*, 1:399–412.
- Tversky, B. and Lee, P. U. (1998). How space structures language. In Freksa et al. (1998), pages 157–176.

- Tversky, B. and Lee, P. U. (1999). Pictorial and verbal tools for conveying routes. In COSIT-99 (1999), pages 51–64.
- van Asselen, M., Fritschy, E., and Postma, A. (2006). The influence of intentional and incidental learning on acquiring spatial knowledge during navigation. *Psych. Res.*, 70(2):151–156.
- van der Zee, E. and Slack, J., editors (2003). *Representing Direction in Language and Space*. Number 1 in Explorations in Language and Space. Oxford Univ. Press.
- Vander Linden, K. and Di Eugenio, B. (1996). A corpus study of negative imperatives in natural language instructions. In *Proc. of 16th Intl. Conf. on Compl. Ling. (COLING-96)*, Copenhagen, Denmark.
- Vander Linden, K. and Martin, J. H. (1995). Expressing local rhetorical relations in instructional text: A case-study of the purpose relation. *Compl. Ling.*, 21(1):29–57.
- Vanetti, E. J. and Allen, G. L. (1988). Communicating environmental knowledge: The impact of verbal and spatial abilities on the production and comprehension of route directions. *Env. & Behavior*, 20:667–682.
- Varges, S. (2005). Spatial descriptions as referring expressions in the MapTask domain. In *Proc. of 10th European Ws. on Nat. Lang. Gen. (ENLG'05)*, Aberdeen, Scotland.
- Verma, V., Estlin, T., Jónsson, A., Pasareanu, C., Simmons, R., and Tso, K. (2005). Plan execution interchange language (PLEXIL) for executable plans and command sequences. In *Proc. of Intl. Symp. on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Munich, Germany.
- Vizard (2006). WorldViz Vizard virtual reality software.
<http://www.worldviz.com/vizard.htm>.

- Ward, S. L., Newcombe, N., and Overton, W. F. (1986). Turn left at the church or three miles north: A study of direction giving and sex differences. *Env. & Behavior*, 18(2):192–213.
- Wauchope, K., Everett, S., Perzanowski, D., and Marsh, E. (1997). Natural language in four spatial interfaces. In *Proc. of 5th Applied Nat. Lang. Proc. Conf. on*, pages 8–11, Washington, DC.
- Webber, B., Badler, N., Di Eugenio, B., Geib, C., Levison, L., and Moore, M. (1995). Instructions, intentions and expectations. *AI*, 73(1–2):253–269. Spec. Issue on “Compl. Res. on Interaction and Agency, Pt. 2”.
- Webber, B. and Di Eugenio, B. (1990). Free adjuncts in natural language instructions. In *Proc. of 13th Intl. Conf. on Compl. Ling. (COLING-90)*, pages 395–400, Helsinki, Finland.
- Weissensteiner, E. and Winter, S. (2004). Landmarks in the communication of route directions. In Egenhofer, M. J., Freksa, C., and Miller, H., editors, *Proc. of GIScience 2004*, volume 3234 of *LNCS*, pages 313–326, Adelphi, MD. Springer.
- Weng, F., Varges, S., Raghunathan, B., Ratiu, F., Pon-Barry, H., Lathrop, B., Zhang, Q., Scheideck, T., Bratt, H., Xu, K., Purver, M., Mishra, R., Raya, M., Peters, S., Meng, Y., Cavedon, L., and Shriberg, L. (2006). CHAT: A Conversational Helper for Automotive Tasks. In *Proc. of Intl. Conf. on Spoken Language Processing (Interspeech/ICSLP)*, pages 1061–1064, Pittsburgh, PA.
- Werner, S., Krieg-Brückner, B., Mallot, H. A., Schweizer, K., and Freksa, C. (1997). Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation. In Jarke, M., Pasedach, K., and Pohl, K., editors, *Informatik 97*, pages 41–50. Berlin, Heidelberg, New York; Springer.
- Winograd, T. (1972). *Understanding Natural Language*. Academic Press, New York.

- Wong, Y. W. and Mooney, R. J. (2006). Learning for semantic parsing with statistical machine translation. In *Proc. of 2006 Human Language Technology Conference - North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL-2006)*, New York City, NY.
- WordNet (2005). WordNet: a lexical database for the English language. <http://wordnet.princeton.edu/>.
- Yeap, W. K. and Jefferies, M. E. (1999). Computing a representation of the local environment. *AI*, 107:265–301.
- Yu, C. and Ballard, D. H. (2004). On the integration of grounding language and learning objects. In *Proc. of 19th Natl. Conf. on AI (AAAI-2004)*, pages 488–493, San Jose, CA.
- Zelek, J. S. (1997). Human-robot interaction with minimal spanning natural language template for autonomous and tele-operated control. In *Proc. of IEEE/RSJ Intl. Conf. on Intelligent Robots & Sys. (IROS-97)*, pages 299–305, Grenoble, France.

Vita

Matthew Tierney MacMahon was born in N. Tarrytown, New York on September 10, 1974, the son of B.J. and Paul MacMahon. Matt attended Marcus High School in Flower Mound, Texas and graduated from the Texas Academy of Mathematics and Science, Denton, Texas in 1993. He earned a Bachelor of Science from the Symbolic Systems Program, an inter-disciplinary cognitive sciences program, at Stanford University in 1997. For three years, Matt worked as an Intelligent Systems Integration Engineer at NASA Johnson Space Center in Houston, Texas, for a contractor. In 2000, Matt entered The Graduate School of The University of Texas at Austin, where in 2002, Matt earned a Master of Science in Engineering degree. During the summers in graduate school, Matt worked for the Naval Research Laboratory in Washington, DC, and NASA Ames Research Center in Moffett Field, CA. In 2003, Matt married Sarah Piper in Austin, Texas. Matt has published papers in the Proceedings of the International Conference on Field and Service Robotics, The National Conference on Artificial Intelligence, The Cognitive Science Society, The IEEE International Conference on Robotics and Automation, The International Conference on Autonomous Agents, The IEEE Aerospace Conference, and The IEEE International Joint Symposia on Intelligence and Systems, as well as articles in AI Magazine, the Journal of Autonomous Agents and Multi-Agent Systems, and Connection Science.

Permanent Address: Matt@MacMahon.org, 12706 Theriot Trail, Austin, Texas.

This dissertation was typeset with $\text{\LaTeX} 2_{\epsilon}$ ¹ by the author.

¹ $\text{\LaTeX} 2_{\epsilon}$ is an extension of \LaTeX . \LaTeX is a collection of macros for \TeX . \TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay and James A. Bednar.